

TEKNILLINEN KORKEAKOULU
Tietotekniikan osasto

Jussi Ranta

**SIMULOINTIMALLIN KÄYTÖN INTEGROITU HALLINTA- JA
ANALYYSIJÄRJESTELMÄ**

**Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi
diplomi-insinöörin tutkintoa varten Espoossa 1.6.2003**

Työn valvoja: Professori Kari Koskinen

Työn ohjaaja: DI Heikki Huotari

Tekijä: Jussi Ranta

Työn nimi: Simulointimallin käytön integroitu hallinta- ja analyysijärjestelmä

Sivumäärä: 89

Päivämäärä: 1.6.2003

Osasto: Tietotekniikan osasto

Professori: Tik-86 Tuotannon tietotekniikka

Työn valvoja: Professori Kari Koskinen

Työn ohjaaja: DI Heikki Huotari, EP-Logistics Oy

Tämä diplomityö on tehty simuloinnin tuotekehityksenä EP-Logistics Oy:ssä. EP-Logistics on logistiikan asiantuntijapalveluihin erikoistunut yritys, joka toimii laitteisto- ja ohjelmistotoimittajista riippumattomana logistiikan suunnittelijana. Simulointia EP-Logistics Oy:ssä on hyödynnetty vuodesta 1983.

Diplomityön jakautuu teorian ja käytännön osan tavoitteisiin.

Teoriaosan tavoite on toimia tulevaisuudessa käytettävänä malliprojektina EP-Logisticsin ohjelmistototeutuksessa. Työn teoriaosa sisältää tapahtumapohjaisen simuloinnin teoriaa, tietokantoihin ja käyttöliittymiin sisältyviä teorioita sekä työssä käytettyjä työkaluja. Teoriaosassa käsitellään myös malliprojektin kulkua ohjelmistototeutuksena ja esitellään olemassa olevia sovelluksia. Teoriaosan oppeja hyödyntämällä tehtiin myös tämän diplomityön simulointimallin työkalusoveltaminen.

Diplomityön soveltavan osan tavoite oli mallintaa henkilöhissijärjestelmälle simulointimalli ja integroida mallille oma käyttöliittymä ja tulosten tallennus yhdeksi sovellustyökaluksi. Tästä muodostuneesta työkalusovelluksesta rakentui simuloidun henkilöhissijärjestelmän käytön hallinta- ja analyysijärjestelmä, jota hyödynnettiin osana kauppakeskus Jumbon sisälogistiikan selvitystä.

Työkalusovelluksen avulla mitoitettava hissijärjestelmä rajattiin koskemaan kauppakeskus Jumbon laajennusosan henkilöliikennettä. Simuloinnin työkalusovelluksen avulla varmistettiin henkilöhissijärjestelmien kapasiteetin riittävyys arvioiduilla henkilöliikennemäärillä. Saatujen analyysien perusteella voitiin tarkemmin mitoittaa tarvittavien hissien lukumäärä ja korikoko.

Diplomityössä tehtyä sovellustyökalua on pystytty myöhemmin hyödyntämään myös muissa tämän diplomityön ulkopuolisissa simuloinnin sovelluskohteissa.

Author: Jussi Ranta

Title of the thesis: Integrated control and analysis tool for simulation usage

Number of pages: 89

Date: 1.6.2003

Department: Department of Computer Science and Engineering

Professorship: Tik-86 Information Technology in Production

Supervisor: Professor Kari Koskinen

Instructor: M.Sc (tech) Heikki Huotari, EP-Logistics Ltd

This thesis has been done in EP-Logistics Ltd as a simulation product development. EP-Logistics is an independent logistics consultant, with no ties to equipment and software suppliers. Simulation has been used in EP-Logistics Ltd since 1983.

The thesis consists of two parts.

In the first part, the theory's goal is to be used as a model manual for future software design. The first part consists of the theory and application of discrete event simulation as well as theories concerning database, interfaces and tools used in this thesis. The first part also introduces the process of the project model's software design as well as existing applications supporting simulation. These theories have been utilized in creating a simulation tool presented in the second part.

The second part of the thesis presents the simulation project of the passenger elevators with an integrated interface and recording database. This integrated application is called the simulation tool which is used as a control and analysis system for simulated passenger elevator. This simulation tool was utilized in the commercial center Jumbo as a part of an operational logistic survey.

Passenger elevators designed with the help of the simulation tool were outlined to concern human traffic in extension the part of the commercial center Jumbo. Passenger elevator capacities, with estimated passenger traffic, were ensured with the help of the simulation tool.

With the simulation tool's results the number and size of the passenger elevators could be measured more accurately than before.

The simulation tool created in this thesis has also been reused in other simulation cases beyond this master thesis.

ALKUSANAT


Tämä diplomityö on tehty EP-Logistics Oy:ssä. Työn valvojana toiminutta professori Kari Koskista kiitän mutkattomasta yhteistyöstä ja ennakkoluulottomasta suhtautumisesta työn aihetta kohtaan.

Työn ohjaajana toiminutta DI Heikki Huotaria haluan kiittää erittäin asiantuntevista kommentteista ja opastuksesta sekä kannustavasta asenteesta työn aikana. Haluan myös kiittää kaikkia niitä henkilöitä EP-Logisticsilla, joilta olen saanut apua työn toteutuksessa.

Kaikkia projektiin osallistuneita kiitän sujuvasta yhteistyöstä projektin aikana. Lisäksi kiitän Kauppakeskus Jumboa avoimesta ja positiivisesta suhtautumisesta diplomityöhöni.

Kaunis kiitos kuuluu myös vanhemmilleni heidän jatkuvasta opiskeluun innostavasta asenteestaan ja kaikesta saadusta tuesta sen aikana. Siskoani Elinaa kiitän työni kieliasuun liittyvistä luonteenomaisen tarkoista kommentteista. Lopuksi haluan osoittaa erityisen lämpimät kiitokset avopuolisolleni Metelle työn aikana saamastani tuesta ja kärsivällisestä suhtautumisesta työn vaatimaan ajankäyttöön.

Espoossa toukokuun 16. päivänä 2003



Jussi Ranta

SISÄLLYSLUETTELO

1	JOHDANTO	5
1.1	TYÖN TAUSTAA	5
1.2	TYÖN TAVOITTEET	5
1.3	TYÖN RAKENNE JA KÄYTETYT METODIT	6
2	TAPAHTUMAPOHJAINEN SIMULOINTI	7
2.1	SIMULOINNIN MÄÄRITELMÄ	7
2.2	SIMULOINNIN MENETELMÄT	8
2.3	SIMULOINNILLA TUTKITTAVAT KOHTEET	10
2.4	SIMULOINNIN TAVOITTEET	11
2.5	SIMULOINNIN SOVELLUSKOHTEET	11
2.5.1	<i>Järjestelmien suunnittelu</i>	12
2.5.2	<i>Järjestelmien kehittäminen</i>	12
2.5.3	<i>Operatiivinen suunnittelu</i>	12
2.5.4	<i>Koulutus</i>	13
2.5.5	<i>Markkinointi</i>	13
2.6	TIETOKANNAT JA KÄYTTÖLIITTYMÄ OSAKSI SIMULOINTIA	13
3	TIETOKANNAT	15
3.1	TIETOKANTAMALLIT	15
3.1.1	<i>Hierarkkinen tietokantamalli</i>	16
3.1.2	<i>Verkkotietokantamalli</i>	17
3.1.3	<i>Relaatiotietokantamalli</i>	18
3.1.4	<i>Uudet tietokantaratkaisut</i>	19
3.2	TIETOKANNAN JA HALLINTAJÄRJESTELMÄN SUUNNITTELU	20
3.2.1	<i>Vaatimusten analysointivaihe</i>	20
3.2.2	<i>Datan muotoiluvaihe</i>	21
3.2.3	<i>Normalisointivaihe</i>	24
3.3	TIETOKANTOJEN INTEGROINTI SIMULOINTIIN	25
4	KÄYTTÖLIITTYMÄ	27
4.1	KÄYTTÄJIEN VAATIMUSMÄÄRITTELYN ANALYSOINTI	27
4.1.1	<i>Toimintaympäristö ja käyttäjät</i>	27
4.1.2	<i>Tarpeiden analysointi</i>	28
4.1.3	<i>käytettävyyystavoitteiden asettaminen</i>	29
4.2	KÄYTTÖLIITTYMÄN SUUNNITTELUPROSESSI	30
4.2.1	<i>Käytettävyyssmenetelmien valinta suunnittelussa</i>	30
4.2.2	<i>Heuristinen suunnittelu</i>	32
4.2.3	<i>Visuaalisuus</i>	35
4.3	KÄYTETTÄVYYSTEKIJÖIDEN ANALYSOINTI	36
4.3.1	<i>Opittavuus</i>	37
4.3.2	<i>Tehokkuus</i>	38
4.3.3	<i>Muistettavuus</i>	38
4.3.4	<i>Virheettömyys</i>	39
4.3.5	<i>Tyytyväisyys</i>	40
4.4	KÄYTTÖLIITTYMÄN INTEGROINTI SIMULOINTIIN JA TIETOKANTOIHIIN	40

5	PROJEKTISSA KÄYTETYT TYÖKALUT.....	42
5.1	YLEISET OHJELMOINTIKIELET	42
5.1.1	SQL.....	42
5.1.2	VBA	43
5.2	OHJELMOINTIOHJELMISTOT	43
5.2.1	ProModel.....	43
5.2.2	Access.....	44
5.3	RAJAPINTA SIMULOINTIIN	44
5.3.1	Active-X.....	44
6	PROJEKTIN TOTEUTUS	45
6.1	PROJEKTIN VAIHEET	45
6.2	ONGELMAN JA TAVOITTEIDEN MÄÄRITTELY	46
6.2.1	Ongelman määrittely	46
6.2.2	Tavoitteiden asettaminen	47
6.2.3	Projektisuunnitelma.....	47
6.3	OHJELMISTON SUUNNITTELU	48
6.4	OHJELMOINTI JA VERIFIOINTI.....	49
6.4.1	Ohjelmointi.....	49
6.4.2	Verifiointi	49
6.5	INTEGROINTI JA VALIDOINTI	49
6.5.1	Integrointi.....	49
6.5.2	Validointi.....	50
6.6	IMPLEMENTOINTI.....	50
6.7	YLLÄPITO.....	50
6.8	TESTAUS	50
6.9	PROJEKTIN AIKATAULUN JA TYÖMÄÄRÄN ARVIOINTI	52
6.10	PROJEKTIN ARVIOINTI.....	52
7	OLEMASSA OLEVIENT JÄRJESTELMIEN TUTKIMINEN JA VERTAILU	53
7.1	INDY-PROJEKTI.....	53
7.2	INDYNET-JÄRJESTELMÄ	55
8	HISSISIMULOINNIN TYÖKALUSOVELLUS.....	57
8.1	TYÖKALUSOVELLUKSEN TAUSTA, TARVE JA TAVOITTEET	57
8.2	SIMULOINTIMALLIN MÄÄRITTELY JA MITTARIT	58
8.2.1	Tekniset parametrit.....	58
8.2.2	Henkilöliikennevirrat.....	59
8.2.3	Ohjausperiaatteet	59
8.2.4	Rajoitukset ja oletukset.....	60
8.2.5	Mittarit	60
8.3	SIMULOINTIMALLIN SUUNNITTELU JA TOTEUTUS	61
8.3.1	Simulointimallin lähtötiedot	61
8.3.2	Simulointimalli	61
8.3.3	Simulointimallin tulokset	62
8.4	TYÖKALUSOVELLUKSEN MÄÄRITTELY JA TAVOITTEET.....	63
8.4.1	Työkalusovelluksen helppo asentaminen	64
8.4.2	Parametrien antaminen	64

8.4.3	<i>Simuloinnin käynnistäminen ja ajojen suorittaminen</i>	64
8.4.4	<i>Tulosten esittäminen</i>	65
8.4.5	<i>Standardianalyysin luonti ja tallennus ajojen vertailua varten</i>	65
8.4.6	<i>Käyttäjäystävällisyys</i>	65
8.5	SOVELLUKSEN SUUNNITTELU JA TOTEUTUS	65
8.5.1	<i>Heuristinen suunnittelu</i>	66
8.5.2	<i>Rajaus</i>	68
8.5.3	<i>Rakenne ja rajapinnat</i>	68
8.5.4	<i>Visuaalisuus</i>	70
8.5.5	<i>Toiminnallisuuden testaaminen</i>	70
8.6	RAPORTOINTI	71
9	TYÖKALUSOVELLUKSEN KÄYTTÖÖNOTTO PROJEKTISSA	73
9.1	PROJEKTIN TAUSTA	73
9.2	ONGELMAN MÄÄRITTELY JA TAVOITTEIDEN ASETTAMINEN	73
9.3	MITOITETTAVA HISSIJÄRJESTELMÄ	74
9.4	TYÖKALUSOVELLUKSEN KÄYTTÖ	74
9.5	TULOKSET JA JOHTOPÄÄTÖKSET	76
9.5.1	<i>Asiakasliikenne ilman liukutasoa</i>	76
9.5.2	<i>Hissien suorituskkyky erilaisilla asiakasmäärillä</i>	76
9.5.3	<i>Suorituskyyvyn vaihteluherkkyys erilaisilla kerrospainotuksilla</i>	79
9.6	YHTEENVETO JOHTOPÄÄTÖKSISTÄ	80
10	YHTEENVETO JA ARVIOINTI TYÖKALUSOVELLUKSESTA	82
10.1	TYÖKALUSOVELLUKSEN ARVIOINTI	82
10.2	HYÖDYT	82
10.3	TYÖKALUSOVELLUKSEN KEHITYSIDEAT JATKOSSA	83
10.4	TYÖKALUSOVELLUKSEN VERTAILU MUIHIN SOVELLUKSIIN	84
11	LÄHDELUETTELO	85
11.1	KIRJALLISET LÄHTEET	85
11.2	INTERNET-LÄHTEET	87
11.3	SUULLISET LÄHTEET	88
12	LIITTEET	89

KÄSITTEET JA LYHENTEET

Hissiryhmä	Yhteisellä ohjausjärjestelmällä varustettu kahden tai useamman hissin ryhmä, johon kuuluvilla hisseillä on yleensä sama nimelliskuorma ja nimellismoisuus sekä samat pysähdystasot.
Lähtöväli	Samasta kerroksesta tapahtuvien lähtöjen keskimääräinen väliaika. Ihmisten keskimääräinen odotusaika on yleensä 0,5-1 kertainen lähtöväliin verrattuna.
Run-time	Käytetyn ohjelman supistettu versio, jossa on vain osa ohjelman toiminnasta käytössä. Usein voidaan käyttää ilman ohjelman lisenssimaksuja.
Täyskoontaohjaus	Välikerroksissa on kaksi kutsupainonappia, joilla matkustaja voi ilmoittaa haluamansa ajosuunnan. Ylöspäin ajava hissi noudattaa vain ylöspäin suuntautuvia kerroskutsuja ja alaspäin ajava hissi alaspäin suuntautuvia kerroskutsuja.
VBA	Visual Basic for Application on ohjelmointikieli Windows-pohjaisille järjestelmille. Sen avulla voi mukauttaa valmisohjelmia ja integroida ne käytössä oleviin tietoihin ja järjestelmiin.

1 JOHDANTO

1.1 Työn taustaa

EP-Logistics Oy on logistiikan asiantuntijapalveluihin erikoistunut yritys, joka toimii laite- ja ohjelmistotoimittajista riippumattomana logistiikan suunnittelija. Osaamis-alueita löytyy monilta logistiikan toimialoilta, kuten teollisuuden, sataman ja liikenteen, julkishallinnon ja kaupan eri sektoreilta. EP-Logistics on osallistunut usean kauppakeskuksen henkilöliikenteen suunnitteluun, joten yrityksellä on ollut tarvetta mitoitustyöhön käytettävällä työkalusovellukselle. Tarpeita vastaava mitoitustyökalu toteutettiin tässä diplomityössä työkalusovelluksena, jota voidaan käyttää hyväksi hissiliikenteen mitoitusprojekteissa.

Olemassa olevien hissivalmistajien mitoitushjeet ja Rakennustietosäätiön RT-ohjekortit on tarkoitettu sovellettavaksi lähinnä toimisto-, virasto- ja liikerakennusten, koulujen, hotellien, sairaaloiden sekä asuinkerrostalojen suunnitteluun. Ostoskeskusten hissimitoitus on hieman erilainen verrattuna toimistorakennusten mitoitukseen suurempien henkilövirtojen vuoksi. Diplomityössä rakennettua ostoskeskuksen hissimitoituksen analysointiin tarkoitettua työkalusovellusta käytettiin Vantaalla sijaitsevassa kauppakeskus Jumbossa, sen laajentaessaan tilojaan. Projektin tarkoituksena oli varmistaa kauppakeskuksen sisäisen liikenteen toimivuus, mikä käsitti sisäisen liikenteen suunnittelun kauppakeskuksen laajennusosassa. Tämä diplomityö on tehty osana EP-Logisticsin simuloinnin kehitystyöprojektia, ja se toimii samalla projektimallina sekä tehtynä pilottiprojektina tulevaisuuden simulointikehityksille.

1.2 Työn tavoitteet

Diplomityön tavoitteet jakautuvat kahteen osaan: teoria- ja käytännönsan tavoitteisiin. Teoriaosan tavoite on toimia käytettävänä malliprojektina EP-Logisticsin ohjelmistototeutuksessa. Teoriaosan oppeja hyödyntämällä tehtiin myös tämän diplomityön simulointimallin työkalusoveltaminen. Diplomityön soveltavan osan tavoite oli saada integroitua simulointimallille oma käyttöliittymä ja tulosten tallennus yhdeksi sovellustyökaluksi. Tässä työssä käytännössä rakennetun työkalusovelluksen tavoitteita käsitellään tarkemmin työn soveltavassa osassa.

1.3 Työn rakenne ja käytetyt metodit

Diplomityö jakautuu kahteen osaan. Ensimmäisessä, teoriaosassa käsitellään tapahtumapohjaisen simuloinnin, tietokantojen ja käyttöliittymän yleistä teoriaa. Lisäksi perehdytään projektin vaiheisiin ja toteutukseen, sekä esitellään olemassa olevia järjestelmiä. Työn toisessa, soveltavassa osassa esitellään työkalusovelluksen toteutusta ja sen soveltamista käytännössä teoriaosassa kuvatun projektinhallinnan keinoin.

Diplomityössä on perehdytty asiantuntijoiden haastatteluin, alan kirjallisuuden, lehdistärtikkeleiden sekä konfrens- ja luentomateriaalien avulla yleiseen teoriaan simuloinnista, tietokannoista ja käyttöliittymästä. Näitä tietoja käytetään hyväksi työn soveltavassa osassa. Lisäksi tutkimuksen aikana on perehdytty muihin olemassa oleviin simuloinnin työkalusovelluksiin. Työn soveltavassa osuudessa simulointityökaluna on käytetty ProModel-simulointiohjelmistoa, tietokantana Access-relaatiotietokantaa, ja käyttöliittymä on rakennettu VBA-ohjelmointikielellä. Eri ohjelmien rajapinnat on yhdistetty Active-X toteutuksella.

2 TAPAHTUMAPOHJAINEN SIMULOINTI

2.1 Simuloinnin määritelmä

Termi "simulointi" on määritelty eri tavoin eri asiayhteyksissä. Simuloinnilla voidaan tarkoittaa esimerkiksi markkinointitarkoituksiin kehitettyä animaatiota, prosessinkehitystyöhön suunniteltua sosiaalista simulointia tai lentokoneohjauksen opetteluun luotua opetussimulaattoria. Laajasti ottaen tietokonesimuloinnille on esitetty esimerkiksi seuraavia määritelmiä:

"Simulointi on numeerinen tekniikka, jossa tietokoneen avulla voidaan suorittaa erilaisia kokeita. Matemaattisilla ja loogisilla malleilla kuvataan monimutkaisen testattavan käyttäytymistä tietyn ajan suhteen." (Naylor 1971, s.2)

"Todellisen järjestelmän loogisen tai matemaattisen mallin rakentaminen, jossa tarkoituksena on tehdä kokeita, joilla pyritään kuvaamaan, selittämään tai ennustamaan todellisen järjestelmän toimintaa." (Hoover & Perry 1989, s.5)

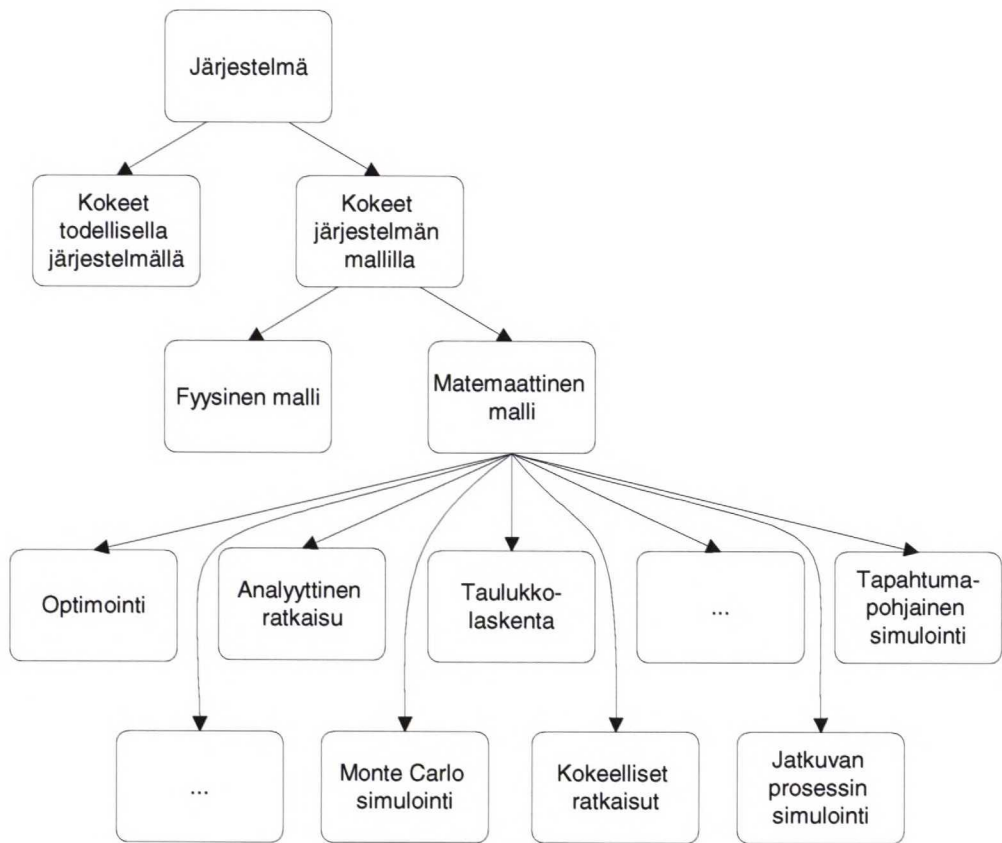
"Todellisen järjestelmän tai prosessin toiminnan imitoimista ajan suhteen." (Banks et al. 1996, s.3)

Määritelmissä kuvataan tietokonesimuloinnin kaksi tärkeintä menetelmää: numeerinen- sekä kokeellinen menetelmä. Simuloinnin voidaan siis ajatella olevan menetelmä, jossa matemaattisen ja loogisen mallin avulla pyritään jäljittelemään todellisen järjestelmän toimintaa jossain olosuhteissa jonkin aikavälin aikana. Simulointia käytetään tiedon hankkimiseen tutkittavasta järjestelmästä ja järjestelmän toiminnan kuvaamiseen, nämä eivät vielä sinällään ratkaise ongelmia vaan vasta simuloinnin antamista tuloksista voidaan tehdä päätelmiä suositeltavista ratkaisuista. (Hannus & Louhekilpi 1981, s7)

2.2 Simuloinnin menetelmät

Simulointi on vain yksi tapa mallintaa rakennettavia reaalimaailman järjestelmiä. Malleilla pyritään tutkimaan todellisten järjestelmien toimintaa ja ratkomaan niissä esiintyviä ongelmia matemaattisia malleja ja tietokoneohjelmistoa hyväksi käyttäen.

Kuvassa 1 on Korpiharjun (2000), Law & Kelton (1991, s.4) pohjalta esitetty jaottelu käytettävistä tutkimusmenetelmistä.



Kuva 1. Järjestelmän tutkimusmenetelmiä (Korpiharju 2000).

Simulointityökalun käytön edut perustuvat siihen, että työkalulla voidaan tutkia monimutkaisia järjestelmiä tekemättä kokeita todellisella järjestelmällä. Etuihin kuuluvat mm. seuraavat (Korpiharju 2000):

- Mallin tilan huomioon ottaminen muuttuvan ajan mukaan keskiarvolaskennan sijaan.

- Simulointi mahdollistaa monimutkaisten satunnaismuuttujien käytön järjestelmien tutkimisessa, mikä ei ole analyttisellä matematiikalla mahdollista.
- Analyttisiin malleihin verrattuna simulointi voi kuvata järjestelmää tarkemmalla yksityiskohtaisuudella ja siihen voidaan joustavasti sisällyttää mitä erilaisempia osia.
- Simuloinnilla voidaan tutkia pitkän ajanjakson tapahtumia lyhyessä ajassa tai vastaavasti lyhyen ajanjakson tapahtumia pitemmällä aikavälillä.
- Simuloinnilla voidaan myös tutkia järjestelmiä, joita ei oikeasti ole olemassa. Perinteisiä menetelmiä käyttämällä tällaisista järjestelmistä ei aina saada riittävää luotettavuutta, minkä vuoksi koko järjestelmän toimivuus voi helposti jäädä tutkimatta.
- Simuloinnissa voidaan ottaa huomioon jonotus, priorisointi ja monimutkainen reititys.

Simulointi ei silti aina välttämättä ole paras vaihtoehto todellisen järjestelmän tutkimiseen. Simulointityökalulla on myös heikkoutensa (Korpiharju 2000, s.25):

- Simulointimallilla suoritettavat ajot antavat vain estimaatteja todellisista tapahtumista. Jos matemaattinen, analyttinen malli pystytään luomaan, se on parempi kuin simulointi.
- Simulointimallit ovat usein kalliita ja aikaavieviä rakentaa. Tämä on kuitenkin suhteellinen asia. Kun kokemus ja osaamisen kasvavat riittävälle tasolle, kustannukset pienenevät.
- Suuret numeromäärät ja animointi luovat suuremman luottamuksen simuloinnin tuloksiin kuin olisi aiheellista.
- Simuloinnin validointi voi olla ongelmallista. Esimerkiksi suunniteltaville uusille kohteille ei ole aina mahdollista löytää kunnon vertailukohtaa varsinkaan kokonaisuutena.

- Simulointi on kokeellinen menetelmä, joka ei suoraan anna vastausta ongelmiin.

Simulointi on hyvä työkalu useimmissa tilanteissa, mutta se ei aina ole paras keino ongelman ratkaisemisessa. Simulointia ei pidä käyttää jos (Banks & Gibson 1997), (Korpiharju 2000):

- Tutkimukselle ei ole riittävän hyvin ja eksaktisti asetettuja tavoitteita.
- Keskittyminen on suunnattu simulointitutkimuksessa pelkästään mallin koodaukseen, ja tutkimuksesta puuttuu tilastollinen analyysi.
- Ei pystytä määrittämään lähtötietoja tai arvioita.
- Verifiointia ja validointia ei voida tehdä.
- Ongelma voidaan ratkaista kokemuksella ilman simulointia.
- Vaihtoehtoisia järjestelmiä pystytään vertailemaan yksittäisten ajojen avulla.
- Simuloinnin kustannukset ylittävät mahdolliset säästöt.

Tässä työssä tarkastellaan tapahtumapohjaista simulointia, jossa mallin tila muuttuu ajan funktiona satunnaisin tai määritellyin aikavälein tapahtuvien muutoksien. Jatkossa simuloinnilla tarkoitetaan nimenomaan tapahtumapohjaista simulointia.

2.3 Simuloinnilla tutkittavat kohteet

Simuloinnilla voidaan tutkia erilaisia mitattavia kohteita, joista esimerkiksi tuotannossa yleisimpiä ovat läpimenoaikojen, tuotantomäärien ja käyttöasteiden sekä varastojen kokoa seuraavat mittarit. Yksittäisten mittareiden lisäksi simulointi antaa laajemman kuvan myös simuloitavan kohteen tilasta kuten esimerkiksi pullonkaulojen muodostumisesta, puskurivarastojen tarpeesta, häiriöiden vaikutuksesta ja järjestelmän toiminnasta uudella ohjaustavalla. Mittareiden ja tutkittavien tilojen avulla simulointi antaa hyvän kuvan monimutkaisesta järjestelmästä, jota voidaan analysoida. On muistettava, ettei simulointi sellaisenaan ole itsetarkoitus, vaan toimii osana pro-

jektia, jonka avulla tutkittavasta kohteesta päätöksen tekee simulointiin ja sen tuloksiin perehtynyt henkilö. (Korpiharju 2000)

2.4 Simuloinnin tavoitteet

Tavoitteet, joita simuloinnille asetetaan, ovat koko simuloinnin lähtökohta. Huonosti esitettyihin kysymyksiin on hankala antaa hyviä vastauksia. Jos tavoitteet ovat epäselviä, ei simulointikaan anna selviä vastauksia. Simulointi on tällöin syytä jättää tekemättä. Tavoitteita, joita simuloinnille voisi esimerkiksi asettaa ovat uuden järjestelmän toimivuuden varmistaminen, käyttöasteiden nostaminen ja läpimenevän tuotantomäärän lisääminen. (Korpiharju 2000) Simuloinnilla tavoitellaan taloudellisen tuloksen parantamista ohjauksen muutoksilla, resurssien säästöillä tai lisäinvestoinnein. Ohjauksen muutoksilla pyritään löytämään parempi tehokkuus ja käyttöasteet eri resursseille. Resurssien säästöillä taas vältetään lisäinvestoinneista, kun asia voidaan hoitaa muulla, halvemmalla tavalla. Lisäinvestoinnilla lisätään pullonkaulana olevan resurssin kapasiteettia, mikä kasvattaa koko järjestelmän tuotantomääriä.

2.5 Simuloinnin sovelluskohteet

Simuloinnin sovelluskohteita on useita ja ne voidaan jakaa erilaisiin ryhmiin. Seuraavassa sovelluskohteet on jaettu käyttötarkoituksen mukaan viiteen ryhmään:

- järjestelmien suunnittelu
- järjestelmien kehittäminen
- operatiivinen suunnittelu
- koulutus
- markkinointi

Kappaleissa 2.5.1 – 2.5.5 on esitelty tarkemmin sovelluskohteet käyttötarkoituksittain. Kappaleessa 2.6 kerrotaan esiteltyjen sovelluskohteiden käyttöliittymän ja tietokannan tarpeista.

2.5.1 Järjestelmien suunnittelu

Simulointimalleja on käytetty menestyksekkäästi uusia järjestelmiä tai laitehankintoja suunniteltaessa. Esimerkiksi ennen kalliiden laitteistohankintojen tekemistä tai uusien järjestelmien käyttöönottoa voidaan simuloinnilla hakea tukea investointipäätöksille, tai vastaavasti säästää kustannuksissa, mikäli simulointi osoittaa suunnitellun laitteistohankinnan turhaksi. Näin voidaan puuttua tilanteeseen ennen kuin kustannuksia laitteistojen osalta on vielä edes syntynyt. Itse käyttöönottovaiheessa tuotannon ylösajoa voidaan nopeuttaa simuloinnilla, joka kertoo jo ennakolta esimerkiksi mahdollisesti syntyvät pullonkaulat ja tarvittavien välivarastojen koot. Simuloitu, nopeampi ylösajo tuo lisää kustannussäästöjä. (Harrell & Tumay 1995, s.7-8)

2.5.2 Järjestelmien kehittäminen

Järjestelmien kehittämisessä simulointi mahdollistaa hyvin joustavan ja edullisen ratkaisun olemassa olevan järjestelmän testaamiseen itse järjestelmää häiritsemättä. (Law, Kelton 1991, s.20) Järjestelmien kehittämisen kannalta simulointia voidaan hyödyntää päätöksenteossa erilaisten ohjaustapojen vertailussa kuten esimerkiksi imuohjausta valittaessa. (Huotari 2000, s.15) Simuloinnilla voidaan myös ratkaista tuotannon ajoituksen ja resurssien allokoinnin ongelmia sekä määrittää muun muassa vaihtoehtoisia tuotantojaksoja, eri toimintamalleja, vuorojärjestelmiä ja töiden prioriteetteja. Näin yrityksen johto pystyy seuraamaan tilannetta ja tekemään sen mukaisia perusteltuja päätöksiä. (Harrell & Tumay 1995, s.7-11)

2.5.3 Operatiivinen suunnittelu

Operatiivisessa suunnittelussa simulointia voidaan käyttää toimintojen suunnitteluun ja niiden aikataulutukseen. Esimerkiksi erilaisten yleisten liikennejärjestelyjen aikataulutusta ja toimintaa voidaan testata toimintaa kuvaavan mallin avulla. Operatiivisessa suunnittelussa on simuloinnin kannalta kuitenkin rajoitteita käytännön tasolla. Simulointia kannattaa käyttää vain tyypilliseen toimintaan, missä satunnaisia tapahtumia, kuten esimerkiksi liikenteen myöhästymisiä ei esiinny. Operatiivisella tasolla syntyvät muutokset voivat olla nopeita ja yllättäviä. Uuden mallin simulointi, simulointiajon tekeminen ja tulosten analysointi pitäisi saada nopeasti valmiiksi eikä tämä aina onnistu helposti ilman asiantuntijan apua. Myös tiedon oikeellisuuden tulkitse-

minen kerätystä tietomäärästä on haastavaa ilman asiantuntemusta. Usein kerätty tieto on puutteellista tai virheellistä. Käytännönkokemus on osoittanut lähtötietojen keräämiseen käytettyjen tietokantojen olevan joko puutteellinen tai virheellisiä simuloitavan tiedon kannalta. Tämä vaatii tietokantojen käsittelijältä korkeaa osaamista lähtötietojen oikeellisuuden tulkitsemisessa. Näistä syistä johtuen simuloinnilla on ollut varsin vähän käyttöä operatiivisessa suunnittelussa. (Korpiharju 2002)

2.5.4 Koulutus

Simuloinnin hyöty koulutuksen kannalta on tarjota selkeä visuaalinen kuva järjestelmän toiminnasta ja antaa samalla kvantitatiivisia tuloksia uuden ratkaisun toimivuudesta myös päätöksentekijöille. (Harrell & Tumay 1995, s.10) Koulutussimulointi auttaa ymmärtämään, miten erilaiset toimintamallit vuorovaikuttavat järjestelmässä ja miten työntekijät voivat järjestelmää operatiivisesti ohjata. Tätä kautta koulutettava näkee myös oman toimintansa vaikutuksen kokonaisuuteen. (Buxton 2000, s.1569-1571).

2.5.5 Markkinointi

Markkinoinnissa uuden idean myyminen asiakkaalle tai yrityksen sisällä on konkreettisempaa visuaalisen ja kvantitatiivisen simuloinnin avulla. Grafiikan kehittyminen on lisännyt mahdollisuutta käyttää simulointia juuri markkinoinnin tukemiseen. (Harrell & Tumay 1995, s.10)

2.6 Tietokannat ja käyttöliittymä osaksi simulointia

Edellä esitetyissä sovelluskohteissa järjestelmien suunnittelu ja kehittäminen ovat osittain simuloinnin kannalta niin sanottuja kertaluonteisia projekteja, joissa tarvitaan ainoastaan simuloinnista saatuja tuloksia, eikä itse simulointiohjelmaa. Rakennettu simulointimalli on kyllä uudelleen käytettävissä, mutta sen käyttö ei yleensä ole jatkuvaa. Tällöin eri vaiheita käsittelee simulointiin perehtynyt asiantuntija, joka ei tarvitse tukea tietokannoilta tai käyttöliittymältä tehdäkseen johtopäätöksiä simuloinnin antamista tuloksista.

Poikkeuksena tästä ovat samanlaiset mitoituskohdeet, jotka ovat helposti parametrisoitavissa, ja jotka voidaan irrottaa tarkastelun ajaksi erilleen tutkittavasta kokonaisuudesta. Esimerkkinä tästä ovat käyttöön suunnitellut kuljettimet, jotka ovat toiminnaltaan yhtenäisiä, mutta eroavat tiettyjen helposti annettavien parametrien suhteen. Esimerkkeinä parametrisoitavista kohteista ovat kuljettimen nopeus ja koko, sekä kuorman suurin kantavuus. Näissä sovelluksissa simulointia voidaan käyttää suunnittelutyökaluna eri laitevalmistajille tai suunnittelijoille sitä varten rakennetun käyttöliittymän ja tietokantojen avulla. Suunnittelutyökalu antaa valmiit standardianalyysit käytetyistä vaihtoehtoista, joiden mukaan mitoitus on järkevä tehdä kulloisellakin määrättyllä kuormalla. Tulevaisuudessa laitevalmistajilla tulee todennäköisesti olemaan kasvavaa tarvetta juuri tämänkaltaisille mitoitustyökaluille.

Koulutus- ja markkinointitilaisuudet voivat olla hyvin esityspainotteisia, jolloin osallistujat seuraavat vain esitystä. Tällöin ei välttämättä tarvita käyttöliittymän tukea simulointiohjelman automaattiseen esittämiseen. Jos esitys taas on interaktiivinen, eli osallistujat pääsevät itse ohjaamaan simuloitua järjestelmää, on käyttöliittymä pakollinen lisätuki eri simulaatioita demonstroitaessa. Tämä lisää myös koulutuksen ja markkinoinnin mielenkiintoa.

Suuri tarve simuloinnin tukemiseen käyttöliittymällä ja tietokannoilla mitoitustyökalun ohella on myös operatiivisessa suunnittelussa. Kappaleessa 2.5.3 mainittu operatiivisen suunnittelun heikkous voidaan korjata käyttöliittymän ja tietokannan liittämällä operatiiviseen simulointiin. Käyttöliittymän avulla käyttäjälle on helpompaa ja nopeampaa tehdä muutoksia simulointiajoihin ilman simulointitekniikan tuntemusta. Myös tulosten tietokantapohjainen vertailu ja analysointi nopeutuvat valmiiksi rakennettujen standardianalyysien avulla. Näin tulosten esittäminen on tehokasta ja niitä on helppo vertailla eri skenaarioita tutkittaessa. Käyttöönotto operatiivisessa suunnittelussa tulee käyttöliittymästä ja tietokannoista rakennetun työkalun ansiosta toimivammaksi. Myöhemmin luvussa 7 esitellään konkreettisia esimerkkejä olemassa olevista ratkaisuksista operatiivisen suunnittelun tueksi.

Seuraavissa luvuissa tarkennetaan erikseen simuloinnin tukemiseen tarvittavia osia: luvussa 3 perehdytään tietokantoihin ja luvussa 4 käsitellään käyttöliittymiä.

3 TIETOKANNAT

Tietokanta on kokoelma loogisesti yhteenliittyvää tietoa, jossa sisältö noudattaa tarkoin määriteltyjä kuvauksia. Tietokanta on suunniteltu ja rakennettu tietyn käyttäjäryhmän tarpeisiin. Tiedonkäsittelyä varten on kehitetty ohjelmisto eli tietokannan hallintajärjestelmä. (Koski 2002, s.6)

Tietokannat jakautuvat kahteen päätyyppiin: käyttötietokantoihin ja analyyttisiin tietokantoihin. Käyttötietokannat ovat dynaamista dataa tallettavia tietokantoja, joissa kerätään, säilytetään ja muokataan ajan tasalla olevia tietoja. Analyyttiset tietokannat ovat puolestaan staattisia tietokantoja, joiden tietoja ei juuri koskaan muuteta, ja jotka kuvastavat vain tiettyä ajankohtaa. (Hernandez 2000, s.3-4) Esimerkiksi simuloinnin lähtötietoina käytetään yleensä analyyttistä tietokantaa. Näin tiettyä ajankohtana kerättyjä tietoja käytetään pohjana simuloitaessa tulevaisuuden tuotantomääriä.

Simuloinnin kannalta oleellista on ymmärtää käytetyn tietokantamallin soveltuvuus simuloinnin lähtöaineistoksi ja soveltavuudesta työssä suunniteltavan tietokantaverron perusteeksi. Olemassa olevien tietokantojen välillä on suuriakin eroja ja eri simuloitavissa kohteissa on vanhahtaviakin tietokantamalleja, joiden vaarat on syytä tietää ennen kuin alkaa simuloida. Sanonta ”roskaa sisään, roskaa ulos” kuvastaa sitä ongelmaa, joka syntyy kun tietokannoista kerättyjen lähtötietojen oikeellisuutta ei ole tarkistettu ennen simulointia.

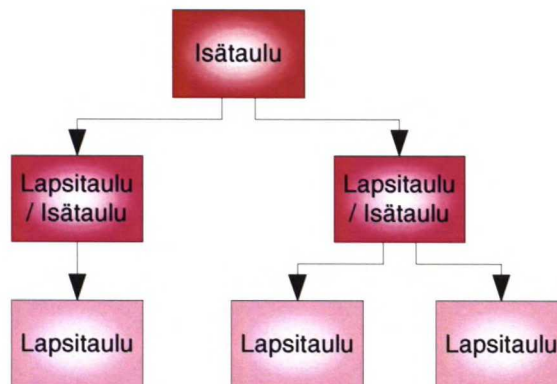
Tässä luvussa tarkastellaan lähemmin eri tietokantojen malleja, niiden etuja ja haittoja, sekä luodaan katsaus tietokantojen hallintajärjestelmän määrittelyn ja suunnittelun periaatteisiin.

3.1 Tietokantamallit

Tietokantamallit jakautuivat yleisesti kolmeen eri malliin: hierarkkiseen-, verkko- ja relaatiotietokantamalliin. Seuraavissa kappaleissa kerrotaan tarkemmin mallien määrittelyä, eduista ja ongelmista, sekä kerrotaan uusista tietokantaratkaisuista.

3.1.1 Hierarkkinen tietokantamalli

Hierarkkisessa tietokantamallissa tiedot on järjestetty hierarkkisesti siten, että yksittäiset taulut kuvautuvat kuten ”puunjuuret” ja muut taulut ovat ikään kuin juuresta lähteviä haaroja. Mallin sisältämiä suhteita kuvataan termien isä/lapsi avulla, missä isätauluun voi liittyä monia lapsitauluja, mutta yksittäiselle lapsitaululle on vain yksi isätaulu. (Hernandez 2000, s.4-5) Kuvassa 2 esitetään kaavio hierarkkisen tietokantamallin rakenteesta.



Kuva 2. Kaavio hierarkkisesta tietokantamallista.

Taulujen yhteys muodostetaan yksiselitteisesti tietueiden fyysisen järjestyksen perusteella. Mallin aloituskohtana on aina juuritaulu, josta kohdetietoon kulkeminen vaatii puun läpikäymistä ja siten hyvää tuntemusta tietokannan rakenteesta. Malli palvelee tehokkaasti vain puurakenteista tiedonhakua (Koski 2002, s.1).

Etuina mallissa ovat: (Hernandez 2000, s.6)

- Nopea tiedon hakeminen, koska taulurakenteet on muodostettu yksiselitteisesti.
- Viite-eheyden automaattinen voimassaolo: lapsitauluilla pitää olla yhteys isätaulun tietueeseen.

Ongelmina mallin käytössä ovat (Hernandez 2000, s.6), (Koski 2002, s.1):

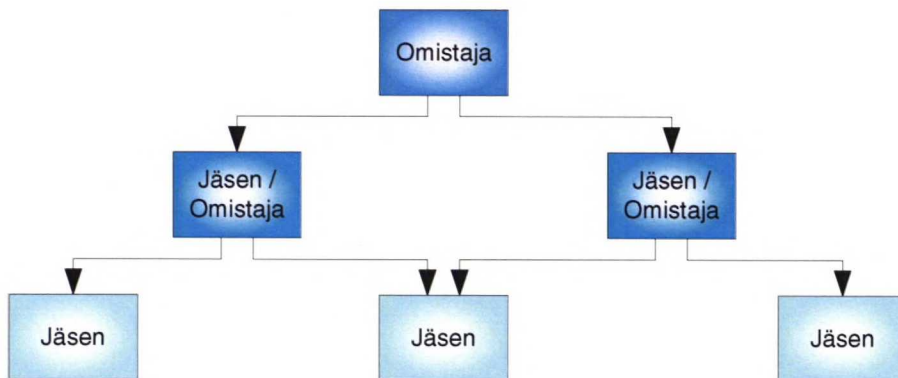
- Tietueen tallennus suoraan lapsitauluun, mikä ei onnistu ilman isätauluun vaadittua jo olemassa olevaa tietuetta.

- Tietoredundanssi eli moninkertainen tietojen tallennus.
- Ylimääräinen data luo päällekkäisyyttä, jolloin tietokannasta voidaan saada virheellistä tietoa.

3.1.2 Verkkotietokantamalli

Verkkotietokantamallia voidaan kuvata samaan tapaan kuin hierarkkista tietokantamalliakin, eli puunjuurina. Siinä taulukoiden yhteys on läpinäkyvä rakenne, joka luodaan joukkorakenteen avulla siten, että toisesta taulusta tulee omistaja ja toisesta jäsen. Omistajataulun tietue voi joukkorakenteen avulla olla yhteydessä moniin jäsentaulun tietueisiin, mutta jäsentaulun tietue on vain yhteydessä yhteen omistajataulun tietueeseen. Lisäksi jäsentaululla on aina olemassa yhteys omistajataulun olemassa olevaan tietueeseen. Kahden eri taulun välille voidaan määritellä ääretön määrä joukkorakenteita ja yksittäinen taulu voidaan liittää kuinka moneen muuhun joukkorakenteeseen tahansa tietokannan muiden taulujen kanssa. (Hernandez 2000, s.9)

Kaavio verkkotietokantamallin rakenteesta kuvassa 3.



Kuva 3. Kaavio verkkotietokantamallista.

Mallin etuja hierarkkiseen tietokantamalliin verrattuna (Hernandez 2000, s.10), (Virtuaali-AMK):

- Tietoa voidaan hakea kulkemalla sopivien joukkorakenteiden kautta ja aloittaa voidaan mistä tahansa taulusta. Tämä lisää hakunopeutta hierarkkiseen malliin verrattuna.
- Joukkorakenteiden avulla vähennetään saman tiedon uudelleen varastointia.

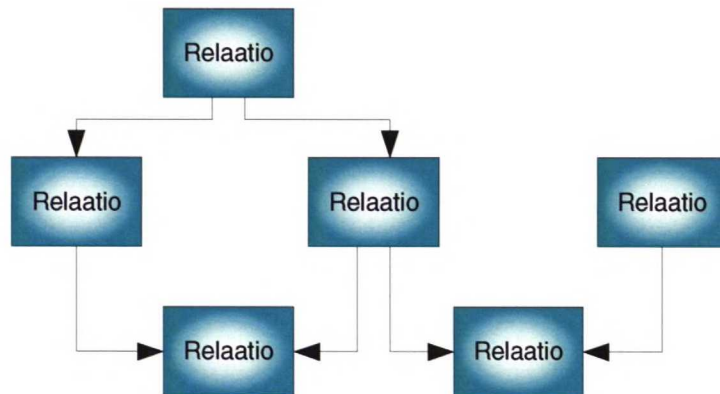
- Voidaan muodostaa monimutkaisempia kyselyjä.

Verkkotietokantamallin ongelmia (Holmström 1987, s.32), (Hernandez 2000, s.10):

- Tietoa tallennetaan sekä tietueisiin että linkkeihin: havainnollisuus kärsii ja tiedon käsittely mutkistuu.
- Tietokannan rakenne on tunnettava hyvin voidakseen liikkua joukkorakenteiden läpi.

3.1.3 Relaatietietokantamalli

E. F. Codd kehitti relaatiotietokantamallin kahden edellä mainitun tietokantamallin ongelmien korjaamiseksi. Relaatietietokantamallissa tietoa talletetaan relaationa eli tauluina. Kaikki taulut muodostuvat kentistä, joiden poikkeavan arvon perusteella tunnistetaan taulujen jokainen tietue. Tämä mahdollistaa tietojen olemassaolon riippumattomuuden fyysisestä tallennusjärjestyksestä. Hierarkki- ja verkkotietokantamallin käyttö edellyttävät tietoa fyysisen rakenteen sijoittelusta, kun taas relaatiotietokantamallissa taulujen välisen yhteyden tunteminen tietoja haettaessa riittää. (Hernandez 2000, s.12) Kaavion rakenteesta on esitetty kuvassa 4.



Kuva 4. Kaavio relaatiotietokantamallista.

Relaatiotietokantojen ylläpito ja käyttö perustuvat SQL-kieleen, joka on rakenteellinen kyselykieli. Tällaisia SQL-kieleen perustuvia järjestelmiä ovat mm. Oracle, Ingres, Informix, SQL Base, SQL Server ja Ms Access (Koski 2002, s.3) (Ullman & Widom 2002, s.19). Lisää SQL-kielestä luvussa 5.1.1.

Relaatiotietokanta tavoittelee tietoriippumattomuutta, tiedon välitettävyyttä ja joukkojen käsittelytavoitteita. *Tietoriippumattomuus* antaa mahdollisuuden tehdä muutoksia tietokantoihin ilman että tietokannan hallintaohjelmia muutetaan ja päin vastoin. *Tiedon välitettävyydellä* tarkoitetaan kaiken tasoisten käyttäjien mahdollisuutta käyttää tietokantoja. *Joukkojen käsittelytavoitteella* mahdollistetaan kykyä ilmaista yhdellä lauseella suuriin tietomääriin kohdistuvia operaatioita. (Koski 2002, s.7)

Relaatiomallilla on useita etuja (Hernandez 2000, s.16):

- Sisäänrakennettu monitasoinen eheys: tietojen oikeellisuuden ja yhteyden varmistaminen sekä puutteellisen tai päällekkäisen tiedon havaitseminen.
- Datan looginen ja fyysinen riippumattomuus tietokantasovelluksista.
- Tietojen yhtäpitävyys ja oikeellisuus.
- Tietojen haun helppous: tieto voidaan hakea mistä tahansa tauluista, joihin sillä on olemassa oleva yhteys.
- Relaatiomalli on yleinen, monenlaisessa käytössä oleva tietokantamalli.

3.1.4 Uudet tietokantaratkaisut

Uudet tietokantaratkaisut, kuten oliotietokannat, hajautetut tietokannat ja monitietokantajärjestelmät pyrkivät käsittelemään perinteisten tietokantojen lisäksi hajautettua tietoa, kuvia, ääntä ja ohjelmia. Etuina uusilla tietokantaratkaisuilla on mutkikkaiden olioiden talletus- ja käsittelymahdollisuus sekä järjestelmän luotettavuuden ja suorituskyvyn lisääminen hajautetuilla tietokannoilla. Haittapuolena ovat kovahkot laitteistovaatimukset ja järjestelmän monimutkaisuuden kasvaminen, jonka ratkaisuksi tarvitaan tietoliikenneverkko. (Koski 2002, s.3)

Tämä työ rajataan jatkossa käsittelemään dynaamista relaatiotietokantamallia. Relaatiotietokantamalli on yleisin, helpoin ja toimivin sekä vähäisellä vaatimuksella luotettavin ratkaisu sovellettavaksi nimenomaan simulointiin. Hierarkki- ja verkkotietokantamallit eivät sovellu muuttuviin ja käsiteltäviin toimintoihin yhtä joustavasti. Relaatiotietokanta ei vaadi käytettävältä laitteistolta suuria resursseja, eikä käyttotar-

koituskaan edellytä uusien tietokantaratkaisujen mukaisia mutkikkaita käsittelymahdollisuuksia. Juuri tämän soveltavuuden vuoksi tässä työssä myöhemmin esiintyvällä ”tietokanta”-termillä tarkoitetaan nimenomaan relaatiotietokantaa, ellei toisin mainita.

3.2 Tietokannan ja hallintajärjestelmän suunnittelu

Tietokannan hallintajärjestelmä on ohjelmisto, joka välittää sovellusohjelmien tietojen käsittelypyynnöt käyttöjärjestelmälle. Tietokannan hallintajärjestelmä tarjoaa yksinkertaisen keinon määrittää ja suorittaa raportointi- ja päivitystehtävät sekä pitää huolta tietokannasta ja sen kuvauksista. (Koski 2002, s.6)

Hallintajärjestelmän suunnitteluun liittyy yleisesti kolme vaihetta (Hernandez 2000, s.28):

- vaatimusten analysointivaihe
- datan muotoiluvaihe
- normalisointivaihe

Kappaleissa 3.2.1 - 3.2.3 kerrotaan tarkemmin suunnittelun eri vaiheista.

3.2.1 Vaatimusten analysointivaihe

Vaatimusten analysointivaiheessa tarkastellaan asikkaana olevaa yritystä, haastatellaan käyttäjiä ja johtoa tulevien tarpeiden analysoimiseksi ja nykyisen järjestelmän kartoittamiseksi. Haastatteluvaiheessa muodostetaan tietokannoille ja hallintajärjestelmille liikesäännöt, joista kerrotaan seuraavassa kappaleessa. (Hernandez 2000, s.28) Hyvällä vaatimusten analysoinnilla voidaan auttaa suunnittelua lisäämään tietokannan luotettavuutta ja eheyttä vähentämällä ylimääräistä dataa tietokannoista. Tavoitteena on saada mahdollisimman yksinkertainen tietokanta ja hallintajärjestelmä vaatimusten toteuttamiseksi. (Ullman & Widom 2002, s.39-40)

3.2.1.1 Liikesäännöt

Liikesääntö on ilmoitus, joka määrää tietokannan eri taulujen asetuksille tai yhteyksille rajoituksia. Liikesäännön rajoitukset perustuvat käsitykseen siitä miten yritys toimii ja käyttää tietojaan. Jokaisella yrityksellä on omat liikesääntönsä, jotka vaikuttavat tietojen, yhteyksien ja tietokannan tuottamien raporttien rakenteeseen. Seuraava lause on esimerkki tyypillisestä liikesäännöstä (Hernandez 2000 s.318-322):

”Laadun takaamiseksi käytämme vain kotimaisia alihankkijoita”

Tämä ilmoitus eli toisin sanoen liikesääntö asettaa ”alihankkija”-kentän ”sallitun arvoalueen” kohdalle rajoituksen, joka sulkee pois ulkomaiset alihankkijat mahdollisista vaihtoehdoista. Suunnittelussa liikesäännöt jaetaan kahteen tyyppiin: *tietokantaan suuntautuneisiin* ja *sovellukseen suuntautuneisiin* liikesääntöihin. Edellä esitetty esimerkkilause kuvaa *tietokantaan suuntautunutta* rajoitusta, joka voidaan ottaa käyttöön loogisen rakenteen suunnittelussa. Tämä tarkoittaa suoraan taulukoiden eri kenttien arvojen rajoittamista. Seuraavaksi tyypillinen esimerkki sovellukseen suuntautuneesta liikesäännöstä (Hernandez 2000 s.318-322):

”Kanta-asiakkaamme saa ostoistaan aina 15 prosentin alennuksen”

Tätä liikesääntöä ei voida enää mielekkäästi ja selkeästi sijoittaa loogisen rakenteen suunnitteluun: ei ole tapaa ilmaista kanta-asiakasstatusta alennuksen selvittämiseksi. Tämän vuoksi liikesääntö otetaan käyttöön tietokantasovelluksena eli fyysisen rakenteen sisällä. (Hernandez 2000, s.318-322)

3.2.2 Datat muotoiluvaihe

Datan muotoiluvaiheessa muotoillaan itse tietokantarakennetta. Muotoilumenetelmillä on olemassa monia malleja, joiden avulla voidaan visuaalisesti esittää tietokantarakenteen eri piirteet kuten taulut, taulujen yhteydet ja yhteyksien ominaisuudet. Tällaisia ovat mm. OMT (Object Modeling Technic), joka perustuu oliotietokannan määrittelyyn ja ER (Entity-Relationship) –malli, joka on yleinen graafiseen kuvaukseen perustuva määrittelymalli. Seuraavassa kappaleessa kerrotaan tarkemmin ER-mallista ja sen ominaisuuksista.

3.2.2.1 Entity Relationship (ER) – malli

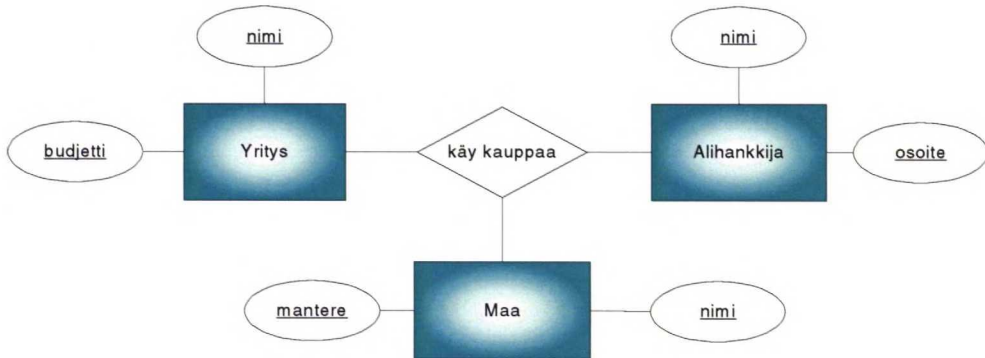
ER-mallittamisessa tarkasteltavaa reaali maailman osaa pyritään hahmottamaan seuraavien peruskäsitteiden avulla (Ullman & Widom 2002, s.24-27) (Mäntylä 1982, s.13-16): yksilö, arvo ja ominaisuus sekä yhteys.

Yksilö on mikä tahansa fyysinen tai abstrakti olio tai asia, josta tietokantaan halutaan tallettaa tietoa. Yksilöt ovat toisista yksilöistä riippumattomia, ”itsenäisiä”, helposti erotettavissa olevia olioita. Esimerkkejä mahdollisista yksilöistä ovat: alihankkija, maa ja yritys. ER-mallin kaaviossa yksilö kuvataan suorakulmiona.

Arvo on jokin yksilöön tai yhteyteen liittyvä seikka, joka ei itse ole yksilö tai yhteys. Se ilmaistaan esimerkiksi luvulla tai merkkijonolla, mutta ei itsessään kerro mitään ilman liittymistä yksilöön tai ilman tietoa arvoa esittävästä asiasta. *Ominaisuus* on funktio, joka tekee arvosta ymmärrettävän ja liittää tämän kohteena olevaan yksilöön tai yhteyteen. Ominaisuus toisin sanoen kuvaa yksilöä ”sen sisältämällä” arvoilla. Esimerkkinä yksilön ”yritys” arvot ”15 000 €” ja ”Firma Oy” ilmoitetaan ominaisuutena ”budjetti” ja ”nimi”. Näin saadaan esimerkin arvoista ymmärrettävä muoto: Yrityksellä, nimeltään Firma Oy on 15 000 € budjetti. ER-mallin kaavioissa ominaisuus kuvataan ovaaleina.

Yhteys on kahden tai useamman yksilön välinen liittymä, joka kuvastaa yksilöiden välistä suhdetta. Esimerkkinä käytettäköön yksilöiden ”yritys” ja ”maa” yhteyttä: yritys ”sijaitsee” maassa. Yhteyden osapuolilla on rooli, joka kertoo yhteyden merkityksen. Edellä mainitussa tapauksessa verbi ”sijaitsee” on esimerkki semanttisesta ilmaisusta, jossa roolit vastaavat lauseenjäseniä (subjekti – objekti). (Mäntylä 1982 s.14) Yksi yhteys voi olla myös monen yksilön välinen, kuten esimerkiksi: yritys ”käy kauppaa” maassa sekä ”käy kauppaa” alihankkijan kanssa. ER-mallin kaavioissa yhteys kuvataan vinoneliönä. Yhteyden tulee myös olla aina jotain tyyppiä, joka kertoo tarkemmin yksilöiden arvojen yhteydestä. Tyyppiä voi olla kolmea erilaista: yhdestä yhteen, yhdestä moneen ja monesta moneen. (Hernandez 2000, s.275) ER-mallin kaaviossa nämä merkitään yllä esitettyssä järjestyksessä 1:1, 1:N, N:N.

ER-malli voidaan esittää havainnollisesti ER-kaavion muodossa edellä esitetyn notaatioin. Se toimii siten suunnittelun havainnollisena apuvälineenä tiedon fyysisessä rakentamisessa. Esimerkki ER-kaaviosta kuvassa 5.



Kuva 5. Esimerkki ER-kaaviosta.

Kuvan 5 kaavio esittää fyysisen tietokannan erilaisia piirteitä. Kaaviossa yksilön voidaan ajatella olevan taulu, johon liitetään ominaisuuksia eli kenttiä. Vastaavasti ominaisuuksien eli kenttien voidaan ajatella olevan tietueita. Taulut ovat keskenään yhteydessä vinoneliöin, mikä kertoo, että tauluilla on fyysinen yhteys toisiinsa. Yhteyden tyypit ilmaisevat taulun yksittäisen tietueen tai monien tietueiden yhteydestä toisen taulun tietueeseen tai tietueisiin.

Kaikille ER-mallin yksilöille on valittava niiden sisältämistä ominaisuuksista pääavain ja mahdollinen viiteavain. Näin varmistetaan ominaisuuksien kunnollinen tunnistaminen ja pakotetaan suunnittelija ymmärtämään yhteyksien paikkansa pitävyys eli eheys. Kuvassa 5 yksilöiden avaimet on valittu ominaisuuksista alleviivausmerkinnällä. *Pääavaimen* valinnan vaatimuksena on ominaisuuden arvojen ”ainutlaatuisuus”. Valinnan pitää kohdistua kenttään, joka määrittää taulun jokaisen tietueen yksiselitteisesti siten, että tiettyyn tietueeseen voidaan aina viitata tarkasti. Pääavain voi muodostua myös usean kentän yhdistelmänä, kunhan jokaista kenttää tarvitaan yksiselitteisen arvon määrittelemisessä. (Hernandez 2000, s.212-220) *Viiteavain* on tietyn taulun pääavainkentästä luotu yhteys toisen taulun samannimiseen kenttään, jossa on suoraan kopioitu sama kenttämääritelmä ja arvoalue. Näin kahden taulun yhteys on määritelty tarkasti yksikäsitteisillä arvoilla ja viittaaminen tiettyyn tietueeseen on täsmällistä. (Hernandez 2000, s.298-301)

Kun alustavat taulukkorakenteet ovat valmiina, ja ER-mallin mukaiset yhteydet on luotu, tietokanta on valmis normalisointivaihetta varten.

3.2.3 Normalisointivaihe

Normalisointivaiheessa suuret taulut hajotetaan pienemmiksi tauluiksi ylimääräisten ja redundanttisten tietojen poistamiseksi, sekä erilaisten tietojen päivitys-, lisäämis- ja poistamisongelmien välttämiseksi. Taulujen rakenteet tarkistetaan eriasteisten normaalimuotojen avulla. Tietokanta on sitä laadukkaampi mitä korkeampiasteinen normaalimuoto on. Normaalimuotoja on viisi sekä lisäksi Boyce-Codd normaalimuoto. Näistä kolmea ensimmäistä normaalimuotoa käytetään yleisesti ja muita teoreettisina erikoistapauksina. (Hernandez 2000, s.30), (Silén 2002)

Seuraavassa käsitellään esimerkein kolmea ensimmäistä normaalimuotoa tarkemmin, koska niiden käyttö on oleellisempaa tietokannoissa. Loppuja erikoistapauksia tarkastellaan lyhyesti ilman esimerkkejä.

1. Ensimmäinen normaalimuoto (1NM)

Tietokanta on ensimmäisessä normaalimuodossa, kun jokaisen taulun kenttä saa vain yhden atomisen arvon. Yhdessä kentässä ei siten saa olla kuin yhtä käytettävää tietoa. Samoin toistuvat ryhmät poistetaan. (Silén 2002) Esimerkki ensimmäisestä normaalimuodosta liitteessä 1.

2. Toinen normaalimuoto (2NM)

Tietokanta on toisessa normaalimuodossa, kun se täyttää ensimmäisen normaalimuodon, ja kun lisäksi taulun kaikki kentät, jotka eivät sisälly pääavaimeen, ovat funktioriippuvaisia koko perusavaimesta eivätkä vain sen osasta. Tauluun pitää siis tallettaa vain sellaista tietoa, joka liittyy pääavaimen määrittelemään asiaan tai kohteeseen. (Koski 2002) Esimerkki toisesta normaalimuodosta liitteessä 1.

3. Kolmas normaalimuoto (3NM)

Tietokanta on kolmannessa normaalimuodossa, kun se on toisessa normaalimuodossa, ja kun pääavaimeen kuulumattomien kenttien välillä ei esiinny täydellistä funktioriippuvuutta. (Koski 2002) Esimerkki kolmannesta normaalimuodosta liitteessä 1.

4. Boyce-Codd normaalimuoto (BCNM)

Boyce-Codd normaalimuoto kieltää määräävän ominaisuuden esiintymisen taulussa, ellei kyseessä ole avain. (Silén 2002)

5. Neljäs normaalimuoto (4NM)

Neljännän normaalimuodon tauluissa ei saa esiintyä kahden tai useamman ominaisuuden moniarvoista riippuvuutta, jos nämä ominaisuudet ovat keskenään riippumattomia. (Silén 2002)

6. Viides normaalimuoto (5NM)

Jos sama informaatio voidaan esittää tauluilla, joiden sarakkeiden lukumäärä on pienempi kuin käytössä oleva, on käytettävä sarakemäärältään pienempiä tauluja. (Silén 2002)

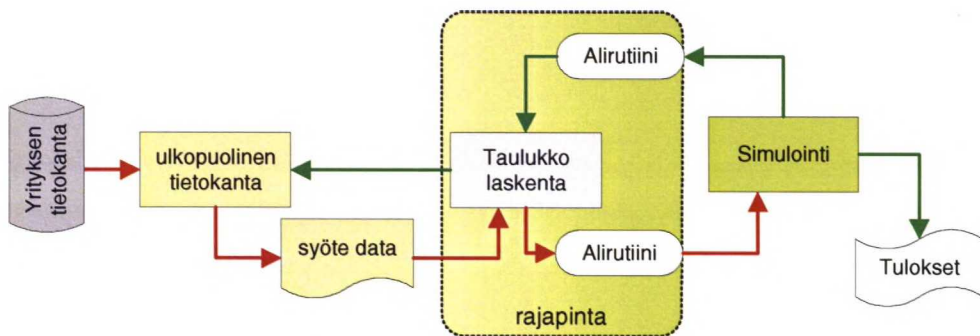
3.3 Tietokantojen integrointi simulointiin

Useimmilla yrityksillä on toiminnastaan jo olemassa olevaa, simuloinnissa hyödynnettävää tietoa omissa tietokannoissaan. Monet simulointiohjelmistot osaavatkin käyttää tietoa suoraan tietokannoista, mutta tietokannoista kerättävä syöte pitää esittää juuri oikealla tavalla simuloinnille, eikä se saa olla virheellistä tulosten oikeellisuuden takaamiseksi. Usein tietokannoista kerättävä tieto on myös hajanaista ja sen koostamiseen syötteeksi tarvitaan ohjelmointitaitoa ja tietoa miten simulointi käyttää tietokantaa. Simuloinnin ja yrityksen tietokantojen välinen kommunikointi on järkevää hoitaa ulkopuolisella tietokannalla, joka työstää yrityksen tietokannat simuloinnin syötteeksi halutuun ehdoin. Syötteiden välitys ohjelmien välillä tapahtuu taulukkolaskentaa ja alirutiineja käyttäen. Alirutiiniohjelmiin voidaan käyttää mitä tahansa ohjelmointikieltä, kuten esimerkiksi Pascalia, Visual Basicia tai Visual C/C++-kieltä.

Tämä mahdollistaa helpon yhdistämisen simulointimallin logiikkaan. (Harrel & Tu-
may 1995, s.64)

Eri taulukkolaskennat ovat yleisesti käytössäolevia ohjelmistoja, joita voidaan hyö-
dyntää simulointiohjelman ja tietokannan välisenä rajapintana yhdessä käytetyn ali-
rutiinin kanssa. Näin itse tietokanta ja simulointimalli ovat fyysisesti erillisiä toisis-
taan, mutta integroituna muodostetun yhteyden kautta. (Nembhard & Kao 2002)

Kokemus on osoittanut simuloinnissa kerättävien lähtötietojen rivikemäärän olevan
usein suurempi kuin mitä normaalit taulukkolaskentaohjelmat pystyvät samanaikai-
sesti käsittelemään. Tämä ongelma muodostaa usein kriittisen heikkouden taulukko-
laskennan suoralle käytölle ulkopuolisena tietokantana. Kuvassa 6 on esimerkkikaa-
vio tietokantojen integroinnista simulointiin.



Kuva 6. Tietokantojen ja simuloinnin integrointi.

4 KÄYTTÖLIITTYMÄ

Käyttöliittymän voidaan ajatella olevan ”ikkuna”, jonka kautta käyttäjä näkee halutun järjestelmän sisällön. ”Ikkuna” mahdollistaa rajallisten ”kehyksien” puitteissa käyttäjän ja käytetyn järjestelmän keskinäisen kommunikoinnin. Kommunikointi tapahtuu käyttäjän antamien komentojen ja syöttötietojen sekä järjestelmän antamien palautteiden kautta. (Bernold 1986, s.3)

Tässä luvussa tarkastellaan lähemmin käyttöliittymiä, niiden määrittelyn ja suunnittelun periaatteita sekä perehdytään käytettävyyteen. Lopussa analysoidaan käyttöliittymää osana tietokantaa ja simulointia.

4.1 Käyttäjien vaatimusmäärittelyn analysointi

4.1.1 Toimintaympäristö ja käyttäjät

Kaikilla sovelluksilla on todellisia käyttäjiä. Käyttäjien joukkoon voi kuulua henkilöitä aina taitavista asiantuntijoista kokemattomiin vasta-alkajiin. Ennen kuin käyttäjistä ja heidän tehtävistään voidaan alkaa luoda kuvauksia suunnittelun tueksi, on yleisellä tasolla selvitettävä, keitä käyttäjät ovat tai tulevat olemaan. Jos sovellusta tuotekehitetään, on käyttäjäjoukko jo selvillä, tai se on selvitettävissä suhteellisen helposti. Vasta käyttäjäjoukon selvityksen jälkeen voidaan tehdä tarkempi luonnehdinta käyttäjien piirteistä, heidän tehtävistään ja toimintaympäristöstään. Käyttäjien tunnistamisen yhteydessä on myös hyvä kuvata, minkälaisessa ympäristössä käyttäjät toimivat. Tätä tietoa voidaan hyödyntää konkreettisesti valittaessa käyttökohdetta, johon suunnittelijat tutustuvat tarpeita analysoidessaan. (Nieminen 2001, s.2)

Käyttäjäjoukkoa pohdittaessa on otettava huomioon myös mahdollisesti sovellusta epäsuorasti käyttävät ihmiset. Heitä voivat olla esimerkiksi sovelluksen antaman paperitulosteen hyödyntäjät, jolloin sovelluksen antaman tulosteen ulkoasuun voi liittyä tietynlaisia vaatimuksia. Erityisenä ryhmänä, joka tulee ottaa huomioon, on sovelluksen ostopäätöksestä vastaava joukko. Vaikka joukon vaatimukset eivät suoraan kohdistu suunniteltavaan käytettävyyteen, voi kyseisillä vaatimuksilla olla sovelluksen markkinoinnissa myyntiargumentillinen arvo. (Nieminen 2001, s.3)

4.1.2 Tarpeiden analysointi

Uuden sovelluksen suunnittelua aloitettaessa on ensin tärkeää suorittaa käyttäjien tarpeiden analysointi. Tämän tutkimiseen saadaan käytännön tietoa osallistumalla todellisiin sovelluksen käyttötilanteisiin. Perehtyminen voidaan tehdä havainnoimalla käyttäjiä ja haastatteleamalla heitä siinä ympäristössä, jossa sovellusta tullaan käyttämään. (Nieminen 2001, s.5)

Ideana on käydä läpi tyypillinen toimintaprosessi ja antaa käyttäjille mahdollisuuden kertoa tehtävistään omin sanoin samalla kun he vastaavat haastattelijan esittämiin kysymyksiin. Oikeanlaisella lähestymisellä saadaan nostetuksi esiin tärkeät, käyttäjien toimintaan vaikuttavat tekijät. Käyttäjiltä voidaan saada tarkennetuilla lisäkysymyksillä arvokkaita tietoja nykyisen sovelluksen parannusehdotuksista sekä työskentelyä haittaavista tekijöistä, joita toivotaan tulevaisuudessa minimoitavan. (Nieminen 2001, s.6-8) Lopullisen työkalusovelluksen ideana on muuttaa tyypillistä toimintaprosessia käyttäjän kannalta helpommaksi ja miellyttävämmäksi.

Tarpeista voidaan tehdä myös käyttöskenaarioita, jotka kuvaavat sovelluksen käyttöä arkielämässä. Skenaario kattaa suunniteltavan sovelluksen keskeisimmät käyttötilanteet ja käyttöön liittyvät mahdolliset ongelmat. Näiden avulla löydetään sovelluksen käytön kannalta olennaisimmat toiminnot. (Nielsen 1993, s.18) Kuvassa 7 on esimerkki mahdollisesta käyttöskenaariosta.

<p>Tavaratalon vahtimestarilla on kiireinen työtahti ja työtila usein hänen sijainnistaan riippuva. Työn toimenkuvaan kuuluu mm. hissien valvonta ja erilaisten teknisten huoltotöiden hoito, sovellusta hyväksi käyttäen.</p> <p>Vahtimestarin pitäisi helposti huomata, kiireestä huolimatta, hissieistä tulevat toimintahäiriövaroitukset. Samalla sovelluksen tulisi antaa huoltotöihin liittyen tarkempia vianmäärittäyksiä, ilman että koko huomio siirtyisi vain niihin, jättäen hissien valvonnan vähemmälle.</p>

Kuva 7. Skenaario vahtimestarin toimenkuvasta.

Skenaariot antavat yhteisen lähtökohdan sovellusta suunnitteleville asiantuntijoille ja toimivat keskustelun pohjana erilaisia priorisointeja rajoituksia mietittäessä. Suunnit-

teluprosessin eri vaiheissa skenaarioita voidaan tarkentaa aina, kun saadaan käyttäjistä lisää tietoa tarkkailun, kyselyjen ja haastattelujen avulla. (Riihiäho 1995, s.111)

4.1.3 käytettävyystavoitteiden asettaminen

Sovelluskehitystyön keskeisen osan muodostaa vaatimusmäärittelyssä toiminnallisuuden määrittely. Toiminnallisuuden määrittely käyttöliittymissä ymmärretään paremmin termillä ”käytettävyystavoitteet”. Käytettävyystavoitteiden asettaminen tulee tehdä yhteistyössä käyttäjien kanssa, jotta tavoitteet muodostuvat realistisiksi ja riittävän laaja-alaisiksi. Suunniteltavan sovelluksen käytettävyystavoitteet voidaan muodostaa toista, vastaavanlaista sovellusta tarkastelemalla. Täysin uudelle järjestelmälle voidaan muodostaa tavoitteet tutkimalla tilanteita, joissa sitä tullaan käyttämään. (Koivunen 1995, s.17)

Sovelluksen vaatimusten selvittämiseen voidaan käyttää apuna erilaisia tehtäväkuvia, asiantuntija-arvioita, standardeja, kyselyjä ja haastatteluja sekä käyttäjän toimintaan ja havainnointiin perustuvia arviointimenetelmiä. Käytettävyydeltään hyviin sovelluksiin pyrittäessä keskeisenä tiedonlähteenä ovat kuvassa 8 esiintyvät käyttöliittymän lopulliset käyttäjät. (Koivunen 1995, s.18)



Kuva 8. Käytettävyystavoitteisiin vaikuttavia tekijöitä (Koivunen 1994, s.18).

Käyttäjien esiintuomien ongelmien ja skenaarioiden pohjalta muodostetaan alustavat reunaehdot sekä tavoitteet sovelluksen käytettävyydelle. (Koivunen 1995, s.20) Käytettävyystekijöille on hyvä määritellä käyttäjien kanssa hyväksymiseen riittävät ta-

voitetasot, joihin pyritään. Käytettävyystavoitteiden määrittelyyn voidaan käyttää erityistä dokumentaatiolomaketta. Lomakkeessa on sarakkeet, joihin sisällytetään määrittelyssä käytetyt, myöhemmin kappaleessa 4.3 tarkemmin esitetyt, käytettävyystekijät ja niiden mittaamiseen valitut mittarit, mittayksiköt ja käytettävyystasot. Perusajatuksena on yhteisen käytettävyyssuunnittelun hyväksyminen mitattavine tekijöineen. Näin tavoitteet ovat selkeät ja kaikkien luettavissa. Dokumentaatiolomakkeella voidaan jo suunnittelun aikana varmistua tavoitteiden saavuttamisesta. (Nieminen 2001, s.10) Taulukossa 1 on Whitesiden et al (1988) esittämästä esimerkki dokumentaatiolomakkeesta.

Taulukko 1. Käytettävyystavoitteiden dokumentaatiolomake Whiteside mukaan.

Käytettävyystekijä	Mittaamiskohde	Mittayksikkö	Huonoin taso	Suunniteltu taso	Paras taso	Nykyinen tilanne
Tyytyväisyys	Ensivaikutelma	Vastaajien määrä	50% neg.	90% pos.	100% pos.	
Opittavuus	Koulutus	arvio ajasta	1 pv	1 h	ei koulutusta	
Muistettavuus	Näppäintoiminnot	Vastaajien määrä	70 %	90 %	100 %	
Virheet	Ohjelman käynnistys	virheiden määrä	4	1	0	
Tehokkuus	Tyypillinen tehtävä	kulunut aika	1 min	30 sek	20 sek	

4.2 Käyttöliittymän suunnitteluprosessi

4.2.1 Käytettävyyssuunnittelun valinta suunnittelussa

Suunnittelun lähtökohtana on määritellä käytettävyydelle rajat, mitä lähdetään kehittämään ja kenelle, sekä mikä on aikataulu ja kehitykseen käytettävissä oleva budjetti. Kaikkia kappaleessa 4.1 esiteltyjä ympäristön sekä henkilöstön tarpeita ja tavoitteita ei aina voida täysipainotteisesti toteuttaa eikä useinkaan ole tarve saada aikaan käytettävyydeltään parasta mahdollista sovellusta. Kun otetaan huomioon sovelluksen elinkaari ja käyttötarve sekä suunnitteluun käytettävissä oleva aika ja raha, ovat rajoitukset aina tapauskohtaisia ja suunnittelijan määriteltävissä.

Kuvassa 9 on Nielseniä mukaillen kuvattu suunnitteluun käytettävien eri menetelmien etuja ja haittoja sekä siihen tarvittavien henkilöiden ja resurssien tarvetta. Kuvassa 9 ”Henkilöiden lkm” ilmoittaa, montako testihenkilöä tarvitaan kunkin menetelmän onnistuneeseen käyttöön suunnitteluvaiheessa.

Menetelmän nimi	Henkilöiden lkm.	Edut	Haitat
Heuristinen suunnittelu	Ei yhtään	Löytää keskeisimmät käytettyysongelmat. Sopii asiantuntijakäyttäjille.	Ei sisällä todellisia käyttäjiä. Yllättävät tarpeet jäävät huomaamatta.
Suorituskyvyn mittaus	Ainakin 10	Mitattavia numerotietoja helppo vertailla.	Ei voida löytää kaikkia keskeisimpiä käytettävyyden ongelmia.
Ääneen ajattelu	3 - 5	Osoittaa käyttäjän harhakäsitykset.	Luonnoton tilanne käyttäjille. Asiantuntijoille hankala ilmaista tekoa sanallisesti.
Havainnointi	3 tai enemmän	Paljastaa käyttäjän oikeat tarpeet ja ehdotukset.	Tapaamiset hankala sopia. Ei kontrolloitua kokeilua.
Kyselyt	Ainakin 30	Subjekttiivinen käyttönäkökulma. Helppo toistaa.	Pilottitutkimus tehtävä ennen kyselyä (vääринymmärryksen estämiseksi).
Haastattelut	5	Joustava, asenteiden ja kokemuksen tunnustelu.	Aikaa vievä. Vaikea analysoida ja vertailla.
Valittu käyttäjäryhmä	Ryhmässä 6 - 9	Välitön palaute ja ryhmäkeskustelua.	Vaikea anylosoida. Vähäinen pätevyys.
Oikea käyttö	Ainakin 20	Löydetään eniten käytettyjä ja käyttämättömiä ominaisuuksia. Voidaan käyttää yhtäjaksoisesti.	Tarvitaan analysointi-ohjelmaa suurille data-määrille. Yksityisyyden loukkaamista.
Käyttäjien palaute	Satoja	Huomataan käyttäjien vaatimat parannukset ja kehitystarpeet.	Tarvitaan erityinen organisaatio palautteiden käsittelyyn.

Kuva 9. Suunnittelun määrittelyssä käytettäviä käytettävyyssuunnittelumenetelmiä etuineen ja haittoineen (Nielsen 1993, s.224).

Kaikki edellä esitetyt menetelmät eivät suoraan ole käytettävissä suunnittelua aloitettaessa, mutta on hyvä pitää mielessä niiden käyttömahdollisuus myöhemmin suunnittelun edetessä. Tyypillisesti projektin rajalliset puitteet sanelevat mitä menetelmää voidaan käyttää. Käytettävyyssuunnittelumenetelmien valinnassa onkin mahdollisuus tehdä eri menetelmien välillä yhdistelmiä, joissa toinen menetelmä korvaa toisen menetelmän puutteet. Esimerkiksi heuristista suunnittelua voidaan täydentää havainnoinnilla, jolloin yllättävätkin tarpeet tulevat otetuksi huomioon. (Nielsen 1993, s.223-226)

Seuraavassa kappaleessa esitellään tarkemmin heuristinen suunnittelu, joka on projektin kustannusten kannalta halvin ja vähiten henkilöiden resursseja kuluttava suunnittelun menetelmä. Se tarjoaa hyvin tehokkaan ja helpon keinon ottaa käytettävyys

huomioon suunniteltaessa pieniä projekteja, joissa käyttöliittymä on vain yksi osa projektia.

4.2.2 Heuristinen suunnittelu

Käyttöliittymän suunnitteluun on kirjallisuudessa lukuisia ohjeita ja yleisiä periaatteita, jotka ovat muodoltaan hyvinkin pitkiä muistilistoja ja joita on usein työlästä käyttää tai muistaa. Ollakseen käytännönläheisiä ja hyödyllisiä, suunnitteluohjeitten tulisi pitää kohtuullisen suppeina ja muistettavina. Molich ja Nielsen ovat laatineet *kymmenen heuristista sääntöä*, jotka kattavat käyttöliittymän tärkeimmät osa-alueet. Lukuisten ohjeiden supistaminen kymmeneen tekee väistämättäkin heuristisista säännöistä hieman väljiä, ja niitä on syytä soveltaan aina asiantuntijan kokemuksen ja sovellusalueen mukaan. Heuristiikan säännöt on esitetty taulukossa 2. (Nielsen 1993, s.19)

Taulukko 2. Kymmenen heuristista sääntöä (Nielsen 1993, s.20).

1	Käytä yksinkertaista ja luonnollista dialogia.
2	Käytä käyttäjien omaa kieltä.
3	Minimoi käyttäjän muistikuorma.
4	Tee käyttöliittymästä kauttaaltaan yhdenmukainen.
5	Anna käyttäjälle palautetta toiminnoista.
6	Anna selkeä poistumistapa eri tiloista ja toiminnoista.
7	Anna käyttäjälle mahdollisuus käyttää oikopolkuja.
8	Anna virhetilanteista selkeät virheilmoitukset.
9	Vältä virhetilanteita.
10	Anna riittävän selkeä apu ja dokumentaatio.

Kappaleissa 4.2.2.1 - 4.2.2.10 luonnehditaan tarkemmin jokaista sääntöä erikseen.

4.2.2.1 Yksinkertainen ja luonnollinen dialogi

Käyttöliittymän tulisi olla mahdollisimman yksinkertainen, eikä siinä saisi olla mitään turhaa, vaan ainoastaan kulloisessakin tehtävässä tarvittavat tiedot. Kaikki harvoin tarvittavat tiedot olisi syytä sijoittaa erilliseen ikkunaan tai alivalikkoon, jolloin ylimääräinen tieto ei heikennä tärkeiden tietojen suhteellista näkyvyyttä. Tiedon pitää esiintyä luonnollisessa ja loogisessa järjestyksessä. (Riihiahho 1995, s.39)

4.2.2.2 Käyttäjien oma kieli

Käyttöliittymän kielen tulisi olla järjestelmän käyttäjän omaa äidinkieltä ja tämän omaa ammattitermistöä. Käyttöliittymän viestit tulisi esittää selkeästi käyttäjän kannalta ajateltuna, välttämättä turhaa hienostelua ja erikoistermejä. Käyttäjälle tuttua sanastoa voi olla vaikea kartoittaa kysymällä sitä suoraan käyttäjältä itseltään. Hyvä tapa luoda sanasto on laatia sovelluksen kehittäjän, käytettävyyssiantuntijoiden ja käyttäjien ehdotusten pohjalta lista sopivista vaihtoehtotermeistä, joista käyttäjä saa valita sopivimmat. Yleisten käyttöliittymien vakiintunut sanasto on muodostunut hyväksi standardiksi erilaisten komentojen ilmaisuun. Esimerkiksi Windows-ympäristön komennot ”Leikkaa” ja ”Liitä” ovat yleisesti tuttuja monelle käyttäjälle. (Nielsen 1993, s.123-129)

4.2.2.3 Käyttäjän muistikuorman minimointi

Tiedon tallentaminen on koneelle helpompaa kuin sen muistaminen tietoa käyttävälle ihmiselle. Koneen tulisi siten huolehtia muistikuormasta ja minimoida se mahdollisimman kevyeksi käyttäjälle. Muistamisen sijaan on hyvä antaa erilaisia vaihtoehtoja, joista käyttäjä voi valita oikean. Esimerkiksi tiedoston nimi on helpompi valita annetusta listasta kuin muistaa se ulkoa juuri oikein kirjoitettuna. Myös erilaisten valikkojen käytöllä vähennetään käyttäjän tarvetta muistaa lukuisia eri komentoja tai vaihtoehtoja. Jos vaihtoehtoja ei voida rajata, on syytä antaa ohjeellinen malli, jonka mukaan käyttäjä osaa tulkita sallitut perusmuodot. Muistikuormaa voidaan lisäksi vähentää noudattamalla koko käyttöliittymässä tiettyä johdonmukaisuutta, josta seuraavassa kohdassa kerrotaan tarkemmin. (Riihiahio 1995, s.40)

4.2.2.4 Yhdenmukaisuus

Käyttäjälle ei saa jäädä epäselväksi, tarkoittavatko erilaiset sanat ja toimenpiteet samaa asiaa, vai johtaako niiden samaksi tulkitseminen virhetilanteisiin. Samantyyppinen käyttöliittymän osien sijoittelu eri ikkunoissa yhdenmukaistaa käsitystä eri toimenpiteistä, joista seuraa aina samantyyppinen tulos. Esimerkiksi ikkunansulkemispainikkeen on tarkoitettava aina samaa ja sijaittava samassa paikassa

eri ikkunoissa. Painikkeen käyttö muuhun tarkoitukseen tai sijoittamiseen eri kohtaan olisi harhaan johtavaa. (Nielsen 1993, s.132-134)

4.2.2.5 *Riittävä palaute*

Palaute on käyttäjälle tärkeää myös virheettömissä tilanteissa, joissa hän on suorittanut valinnan tai komennon oikein. Usein välitön palaute on komennosta seuraava toimenpide, jonka käyttäjä havaitsee järjestelmän hyväksymänä syötteenä ilman viivettä. Viiveettömän ja alle 1 sekunnin kestävän tuloksen esittämiseen ei erillistä palautetta tarvita, mutta tätä pitempään kestävästä toimenpiteistä on hyvä antaa erillinen palaute, joka kertoo toimenpiteen kestosta joko ajanyksikkönä tai toimenpiteen edistymisen myötä täyttyvällä palkilla. Palaute tehostaa aina oppimista ja antaa tunteen järjestelmän hallinnasta. (Riihiahho 1995, s.41)

4.2.2.6 *Selkeä poistumistapa eri tiloista ja toiminnoista*

Käyttöliittymässä on aina oltava näkyvä poispääsy eri tilanteista, jolloin käyttäjälle jää mahdollisuus korjata vahingossa tai kokeilemalla syntynyt toiminto tarvitsematta käydä läpi koko dialogia. Paluumahdollisuus edelliseen tilaan voidaan antaa peruuta-painikkeella, tai käyttäjälle voidaan tarjota poistumistapa koko järjestelmästä. Jos poispääsyä ei ole mahdollista tarjota, on se ilmoitettava käyttäjälle etukäteen ja kysyttävä vielä varmennus toiminnalle, ennen kuin toiminta toteutetaan. (Riihiahho 1995, s.42)

4.2.2.7 *Oikopolut*

Sovellusta pitkään käyttänyt ja sen peruspiirteet osaava käyttäjä haluaa usein tehostaa työskentelyään. Pitkät monisanaiset dialogit, jotka alussa auttavat ohjelmiston opettelussa, voivat myöhemmin osoittautua hitaiksi käyttää. Kokeneelle käyttäjälle tulisikin tarjota vaihtoehtoisia, nopeampia tapoja suorittaa toiminnot oikopolkuja käyttäen. Oikopolut eivät siis koskaan saa olla ainoa tapa suorittaa tehtävä, vaan juuri vaihtoehtoinen apuväline kyseiselle toiminnolle. Tyypillisiä oikopolkuja tarjoavat pikanäppäimet, komentojen lyhenteet ja toimintonäppäimet. Oikopolut on hyvä esittää myös sovelluksessa eikä vain ohjekirjoissa. Näin niiden käyttöönottoa voidaan tehostaa. (Nielsen 1993, s.139-142)

4.2.2.8 *Selkeät virheilmoitukset*

Virhetilanteiden syntyessä tulisi käyttöliittymän antaa mahdollisimman selkeä ja rakentava ilmoitus, jossa kerrotaan tarkasti virheen laadusta ja ohjataan käyttäjää mahdollisesti korjaamaan virhe. Suunnittelijalle helposti avautuva koodirivin virheilmoitus ei tee virheitä selväksi ohjelmaa käyttävälle henkilölle, vaan virheilmoitus tulee esittää selkeästi käyttäjän omalla kielellä. (Riihiahho 1995, s.43)

4.2.2.9 *Virheiden estäminen*

Virheilmoituksia parempi vaihtoehto on tilanne, jossa itse virheet estetään. Ne voidaan estää antamalla esimerkiksi vaihtoehtoja, joista valita, jolloin kirjoitusvirheiden mahdollisuus häviää. Erilaisia virheitä voidaan myös estää pyytämällä käyttäjältä varmistus, ennen kuin esimerkiksi tiedosto lopullisesti tuhoetaan. (Riihiahho 1995, s.44)

Käyttöliittymän yhdenmukaistaminen ehkäisee virheellisiä painalluksia tapahtumasta. Erilaisilla äänillä tai väreillä voidaan myös korostaa virheen mahdollisuutta. Esimerkkinä tästä mainittakoon MS Wordin oikeinkirjoitustoiminto, jossa virheellisen sanan alle ilmestyy sahalaitaviiva osoittamaan mahdollisesta kirjoitusvirheestä.

4.2.2.10 *Selkeä apu ja dokumentaatio*

Ihanteellista käyttöliittymää pitäisi pystyä käyttämään ilman erillisiä ohjeita, mutta tähän ei useinkaan päästä. Sovelluksen tulisi tällöin tarjota erityyppistä tukea sen käytölle. Sovelluksella olisi hyvä olla sisäinen digitaalisessa muodossa oleva ohjeistus sekä paperinen ohjekirja, josta asiat löytyvät jäsennellysti ja helposti. Ohjeiden tulee olla myös mahdollisimman itsenäisiä, eivätkä ne saa nojautua aiemmin esitettyyn tietoon. (Nielsen 1993, s.148-150)

4.2.3 *Visuaalisuus*

Syöttörivien, painikkeiden, tekstien ja eri asioiden sijoittelu ja ryhmittely käyttäjälle avautuvissa ikkunoissaan ovat käyttöliittymän visuaalisuutta. Kuvakkeet toimivat ymmärrettävän tekstin lisänä, metaforisena kielenä, jonka avulla käyttä-

jälle voidaan ilmaista suurempia kokonaisuuksia ja toimintoja esimerkiksi pelkästään yhdellä vertauskuvallisella ja ymmärrettävällä kuvakkeella. Oikeanlainen graafinen sommittelu helpottaa toimintojen löytämistä ja yhdenmukaistaa sovellusta.

Hyvän visuaalisuuden tavoitteena on saada käyttäjän huomio kiinnittymään kuskakin toiminnossa keskeisiin kohtiin. Yhdellä kertaa ei ole suotavaa esittää käyttäjälle useita tietoja, koska jokainen ylimääräinen tieto kilpailee näkyvyydestä tarpeellisen tiedon kanssa. Samaten koko käyttöliittymän visuaalisessa suunnittelussa värien tulisi olla maltillisia ja niiden käytössä tulisi tyytyä vain muutamaan perusväriin. Värit ovat silti suuri apu erilaisten osien ryhmittelemisessä, samoin kuin ovat osien muotoilu ja koko sekä esimerkiksi kirjasinlaatu.

Yhteenkuuluvuutta voidaan korostaa rajaamalla ryhmiä tai yhdistämällä ryhmien osia viivoilla. Liikkuvat, välkkyvät, isot tai värikkäät osat kiinnittävät käyttäjän huomion aivan eri tavalla kuin taustalla olevat osat. Näiden käyttöä onkin syytä harkita korostuskeinona. Kun tiettyyn tehtävään tarvittavat osat on ryhmitelty hyvin, käyttäjä löytää nopeasti tarvitsemansa osat. (Riihiaho 1995, s.39-42 ja Koivunen 1994, s.42)

Visuaaliseen suunnitteluun ei ole olemassa suoranaisia standardimalleja, mutta edellä esiteltyjä visuaalisuuden peruseriaatteita noudattaen on sovelluksesta mahdollista luoda tasapainoinen ja harmoninen kokonaisuus, joka antaa käyttäjälleen luotettavuuden tunteen ja rauhoittaa työympäristöä.

4.3 Käytettävyystekijöiden analysointi

Käytettävyys kuvaa käyttöliittymän laatua, ja sitä voidaan mitata epäsuorasti käytettävyysominaisuuksien avulla. Käytettävyys on siis kuvaus sovelluksen sopivuudesta tehtäväkokonaisuuden suorittamiseen tietyssä ympäristössä. Sovellusten käytettävyyteen voidaan vaikuttaa hallitusti ja perustellusti käytettävyyssuunnittelulla. Käytettävyyssuunnittelussa etsitään vastauksia mm. seuraaviin kysymyksiin (Koivunen & Nieminen, 1995 s.12):

- Millaisia tehtäviä sovelluksen pitäisi tukea?

- Millaisia vaatimuksia käyttäjät ja käyttöympäristö asettavat sovellukselle?
- Miten nopeasti käyttäjä oppii sovelluksen käytön?
- Onko käyttö riittävän sujuvaa ja vaivatonta?
- Millaisia virheitä käytössä tapahtuu ja kuinka käyttäjä selviää niistä?
- Kuinka paljon sovelluksen käyttö kuormittaa käyttäjää?

Jakob Nielsen on tiivistänyt käytettävyydelle esitetyt kysymykset viiteen eri käytettävyyssominaisuuteen, joita voidaan mitata laatua arvioitaessa (Nielsen 1993, s.26):

- Opittavuus: sovelluksen nopea ja helppo oppiminen.
- Tehokkuus: opitun sovelluksen tuottava ja tehokas hyödyntäminen.
- Muistettavuus: pitemmän tauon jälkeinen käyttö ilman uudestaan opettelua.
- Virheettömyys: virheiden määrän ja laadun minimoiminen.
- Tyytyväisyys: sovelluksen käytön positiivinen subjektiivinen kokemus.

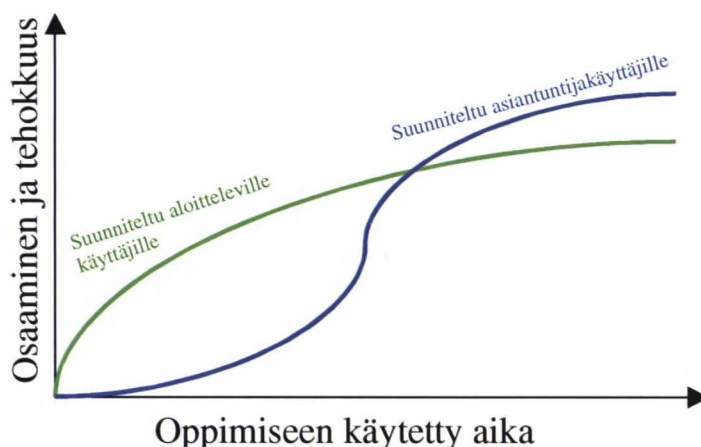
Kappaleissa 4.3.1 – 4.3.5 kerrotaan tarkemmin käytettävyyssominaisuuksista ja niiden mitattavuudesta.

4.3.1 Opittavuus

Opittavuus on tietyllä tavalla oleellisin ja tärkein käytettävyyssominaisuus, koska ensimmäinen asia, jonka käyttäjä kohtaa, on juuri uuden järjestelmän käytön opettelu. Opittavuudella mitataan usein oppimisen nopeutta ja helppoutta. Eri järjestelmille voidaan hyväksyä erimittaisia oppimisaikoja riippuen siitä, millaisille käyttäjille ne on tarkoitettu. Asiantuntijakäyttäjille eli käyttäjille, jotka käyttävät aktiivisesti tietoteknisiä järjestelmiä, voidaan varata enemmän aikaa opetteluun kuin aloitteleville käyttäjille, joilla ei ole tietoteknisiä valmiuksia. Tämän voi perustella asiantuntijakäyttäjille sovelluksen tehokkuuden ja nopeuden tarpeella, missä tehokkaaseen työkentelyyn tarvittavat erikoiskomennot ja räätälöitävyys saattavat lisätä oppimisaikaa.

Asiantuntijakäyttäjät eivät silti välttämättä opi sovellusta perin pohjin ja voivat siten olla aloittelevia käyttäjiä jollakin tietyillä osa-alueilla. (Nielsen 1993 s.28-30)

Kuvassa 10 on Nielsenin esittämä oppimiskäyrä oppimisajan suhteesta tehokkuuteen. Siinä aloitteleville käyttäjille suunniteltu järjestelmä on helppo oppia, mutta ei niin tehokas käyttää. Asiantuntijoille suunniteltu järjestelmä on taas vaikeampi oppia, mutta oppimisen loppuvaiheessa työtehokkuus on selvästi parempi kuin aloitteleville suunnitellussa sovelluksessa.



Kuva 10. Oppimiskäyrä eri käyttäjille (Nielsen 1993 s.28).

4.3.2 Tehokkuus

Tehokkuudella tarkoitetaan käytettävyyden näkökulmasta tasoa, jolle järjestelmän käytön nopeus sijoittuu, kun se on opittu hyvin. Tehokkuuden absoluuttista tasoa on vaikea määrittellä, mutta sen määrittäminen voi perustua esimerkiksi asiantuntijakäyttäjien hyväksymään työskentelyn nopeuden tasoon. Työskentelyn nopeus voi olla siten keskiarvo mittaustuloksista, tai arvo, jonka tietty prosenttimäärä käyttäjistä alittaa. Kuvassa 10 nähdään maksimitehokkuuden saavuttaminen käyrän tasoittumisena. Suunnittelussa sovelluksen tehokkuutta voidaan analysoida antamalla testikäyttäjille työtehtävä, joka heidän pitää suorittaa tietyssä ajassa. (Nielsen 1993 s.30-31)

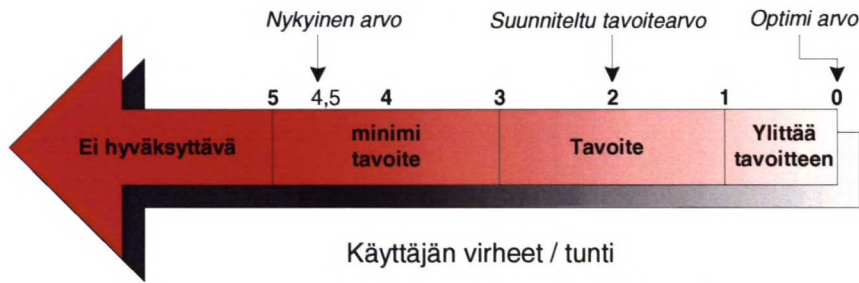
4.3.3 Muistettavuus

Muistettavuus liittyy sovellusten satunnaiseen käyttöön. Toisin sanoen sellaisten sovellusten käyttöön, joita tarvitaan harvoin. Muistettavuus liittyy siten siihen, miten

helposti toimintojen termien ja graafisten merkkien sisältö on palautettavissa mieleen sen jälkeen, kun järjestelmä on kerran opittu. Helppo opittavuus voi myös suoraan vaikuttaa muistettavuuteen, mikäli sovelluksesta on helppo luoda itselleen sisäinen malli. Suunnittelussa sovelluksen oppineita testikäyttäjiä voidaan pyytää käyttämään sovellusta uudestaan pitemmän käyttötaun jälkeen ja mitata aikaa, kuinka nopeasti he muistavat aiemmin oppimaansa. (Koivunen & Nieminen 1995 s.23-24)

4.3.4 Virheettömyys

Sovelluksen tulisi olla selkeä ja johdonmukainen siten, että käyttäjät voivat joustavasti ja hallitusti edetä työtehtävissään joutumatta virhetilanteisiin. Virhetilanteiden syntyessä on myös tärkeää hallita virheistä selviytyminen selkeällä ohjeistuksella, joko suoraan sovelluksen antamien ohjeiden mukaan tai ohjekirjaa hyväksikäyttäen. Sovelluksen suunnittelussa virheiden laatua voidaan myös seurata ja suurimmat virheet, joista seuraa työtehtävän lopullinen keskeytys, on pyrittävä poistamaan kokonaan. Osa virheistä on pienempiä, lähinnä pelkästään aikaa vieviä tapahtumia, joista pystyy silti helposti jatkamaan tehtävää. Kuvassa 11 on esitetty Nielsenin esimerkki tavoitesuorasta, jonka avulla voidaan mitata sovelluksen käytettävyyttä virheiden näkökulmasta. (Nielsen 1993 s.32-33)



Kuva 11. Virheettömyyden tavoitesuora (Nielsen 1993 s.81).

Kuvassa 11 on virheettömyyden tavoitteelle asetettu minimiraja (virheiden lkm/tunti), mitä hyväksyttävä sovellus ei saa ylittää. Päättavoitteena on aina silti optimiarvoon pyrkiminen.

4.3.5 Tyytyväisyys

Tyytyväisyys on käyttäjäkohtainen kokemus käytetystä sovelluksesta. Tyytyväisyys motivoi käyttäjää opettelemaan sovelluksen käytön, kun taas tyytymättömyys estää tämän. Tyytyväisyydellä on myös positiivinen vaikutus käyttäjän tehokkuuteen ja sovelluksen muistettavuuteen. Tyytyväisyys on selvitettävissä suoraan haastattelemalla tai käyttämällä hyväksi erilaisia kyselyjä, missä käyttäjä itse antaa palautetta käytettävyydestä esimerkiksi konkreettisesti mitattavin arvosanoin. (Nielsen 1993, s.33-37)

4.4 Käyttöliittymän integrointi simulointiin ja tietokantoihin

Simulointi on työkalu, jonka menestyksellinen käyttö vaatii syvällistä asiantuntija-osaamista. Simuloinnin asiantuntija on näin ollen usein myös työkalun käyttäjä ja toimii työkalun välityksellä ongelmanratkaisijana simuloinnin tuloksia tarvitsevalle henkilölle. Asiantuntijan voidaan siten ajatella olevan ikään kuin ”käyttöliittymänä” ohjelman ja sitä tarvitsevien henkilöiden välillä. Tällainen asiantuntijan välittämä, ohjelmiston tuloksista kertova tieto on usein riittävä yksittäisissä simulointiprojekteissa. Tällaiset projektit ovat kertaluonteisia, tiettyä asiaa kertaalleen tutkivia simulointitarkasteluja, joissa tulokset esitetään raporttina itse tulosten käyttöönottajalle.

Siirryttäessä kertaluonteisesta simuloinnista saman kohteen jatkuvampaan simulointitarpeeseen, kuten operatiiviseen ohjaukseen, on syytä korvata asiantuntija simulointiin rakennetulla käyttöliittymällä. Käyttöliittymän merkitys korostuu siirryttäessä simuloinnin asiantuntijoista aina konkreettisen tason ohjauksesta vastaaviin käyttäjiin, joilla ei ole laajaa simuloinnin asiantuntemusta. (Odhabi et al 1998)

Erilaiset simulointiohjelmistot tukevat taulukkolaskentapohjaisten käyttöliittymien käyttöä, joissa parametrien muuttaminen tapahtuu suoraan eri taulukkolaskennan solujen tietoja muuttamalla. Taulukkolaskennan käyttöä käyttöliittymänä tukee helppo toistettavuus pitemmille syötesarjoille, joissa sama arvo voidaan kopioida annetun kaavan mukaisesti monille soluille samanaikaisesti. Kuitenkin tämän tarve on varsin vähäistä. Usein käytetty tapa onkin vain yksittäisten parametrien muutos eri simuloinneilla suoritetuissa ajoissa, joiden mukaan tietty lähtötieto on vaihdettavissa sellaisenaan, ilman syötesarjojen kopioimista. Esimerkkinä tästä on yleisten kulkuneu-

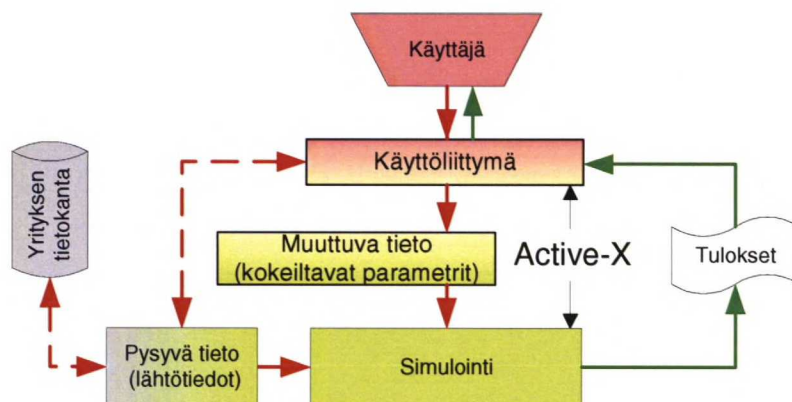
vojen vuoroliikenteen kesä- ja talviaikataulujen vaihto yhdellä parametrilla, jolloin simulointiohjelma ymmärtää käyttäjäkokonaan eri lähtöaikatauluja.

Eri taulukkolaskentaohjelmien käyttö simuloinnin käyttöliittymänä on uusien toimisto-ohjelmien päivitysten vuoksi muodostunut ongelmallisemmaksi kuin Visual Basic eli VBA-tekniikalla (ks. kappale 5.1.2) tehdyt käyttöliittymät. VBA-tekniikka pysyy vakaammin yhdenmukaisena koneiden ohjelmia päivittäessä kuin muut herkemmin muuttuvat perustoimistotyökalut kuten esimerkiksi Excel. (Heilala 2002)

Taulukkolaskentaa monipuolisemmin toimii erillinen VBA-tekniikalla toteutettu tietokantasovellus, jota voidaan käyttöliittymän kautta helposti käsitellä tarvittavin SQL-kyselyin (ks. kappale 5.1.1). Tällä saavutetaan (etuna taulukkolaskentaan) tulostietojen helppo tallentaminen, lisääminen ja muuttaminen erilaisia tarkasteluja varten. SQL-kyselyillä voidaan helposti muuttaa myös harvemmin vaihtuvia lähtötietoja, jotka kerätään yrityksen tietokannasta.

VBA-tekniikan etuna Windows-ympäristössä on käyttää Active-X -toimintoa suoraan ohjausrajapintana käyttöliittymän ja simulointiohjelman välillä, jolloin se välittää käyttöliittymällä annettuja tietoja simuloinnille ja päinvastoin.

Kuvassa 12 on esimerkkikaavio käyttöliittymän integroinnista simulointiin ja tietokantoihin.



Kuva 12. Käyttöliittymän integrointi simulointiin ja tietokantoihin.

5 PROJEKTISSA KÄYTETYT TYÖKALUT

Tässä luvussa tullaan esittelemään lyhyesti diplomityön soveltavassa osassa käytettyjä työkaluja.

5.1 Yleiset ohjelmointikielet

5.1.1 SQL

Structured Query Language eli SQL on tietokantajärjestelmän standardoitu tietokantakieli, jonka tarkoituksena on antaa käyttäjälleen mahdollisuus tiedonmuokkaukseen ja relaattiorakenteen määrittelyyn sekä tiedonhallintaan, kuten tiedon lisäämiseen tai poistamiseen tietokannasta. (Ullman & Widom 2002, s.239)

SQL on erikoistarkoitukseen kehitetty kieli, eli se soveltuu ainoastaan tietokantojen hallintaan, ei sovellusohjelmointiin. SQL:n tarkoitus onkin suorittaa mahdollisimman helposti tietokannoista erilaisia kyselyjä ja tehdä muutoksia, joilla saadaan raaka data hyödynnettäväksi informaatioksi. Kielen käyttö voidaan jakaa vuorovaikutteiseen, upotettuun ja dynaamiseen osaan. Vuorovaikutteinen SQL tarkoittaa SQL-tulkin välityksellä tapahtuvaa ”suoraa” tietokannan muutosta ja kyselyjen tekoa. Upottaminen tarkoittaa SQL-lauseiden lisäämistä johonkin rakenteelliseen ohjelmointikieleen kuten esimerkiksi C++:aan, Visual basiciin tai Perliin. Dynaamista SQL:a käytetään usein sovelluskehittimissä, jossa SQL-käsky muodostetaan dynaamisesti ohjelmassa ja lähetetään tietokantajärjestelmälle käännettäväksi ja suoritettavaksi. (Koski 2002, s.15)

SQL:n etuna on standardoitunut tietokantakieli, jota on helppo käyttää. Puutteena on standardin ominaisuuksien riittämättömyys tietokantajärjestelmien suunnittelijoille, minkä vuoksi SQL:n käytölle on kehitetty lisälaajennuksia. Laajennukset mahdollistavat standardiin kuulumattomien ominaisuuksien, kuten standardoimattomien tietotyyppien sekä erilaisten herättimien käytön. (Herättimillä voidaan käynnistää automaattisesti haluttuja operaatioita.)

5.1.2 VBA

Visual Basic for Applications eli VBA on Microsoftin luoma ohjelmointikieli Windows-pohjaisille järjestelmille. Sen avulla voidaan mukauttaa ja integroida valmisohjelmia käytössä oleviin tietoihin ja järjestelmiin. VBA on muodostunut Microsoft Visual Basiciin (VB) sovelluskehityksen tuloksena. VBA antaa mahdollisuuden mukauttaa eri valmisohjelmia vastaamaan käyttäjän omia erityistarpeita. VBA sisältää myös mukautettujen valintaikkunoiden luomiseen käytettävän Microsoft Forms -tuen sekä Active-X-komponentteja, joiden avulla voidaan luoda käyttöliittymä sekä rajapinta eri ohjelmien välille. (Microsoft 2002)

5.2 Ohjelmointiohjelmistot

5.2.1 ProModel

ProModel on oliosuuntautunut, Windows-pohjainen simulointiohjelmisto, joka on erityisesti kehitetty tuotanto- ja logististen järjestelmien mallintamiseen. Prosessien mallintamiseen on ProModelissa käytettävissä oma, varsin joustava ohjelmointikieli, johon voidaan linkittää myös ulkoisilla ohjelmointikielillä ohjelmoituja rutiineja. ProModel voi kommunikoida myös muiden ohjelmien kanssa käyttäen hyväkseen Active-X -rajapintaa. ProModel pystyy suoraan käyttämään ja kirjoittamaan tietoja tekstitiedostoon tai Excel-taulukkolaskentaan. (ProModel 2003)

Simulointiajossa ProModel kerää mallinnetuista suureista tilastotietoja ja näyttää mallinnuksen animaationa. Simulointiajon jälkeen tuloksia voidaan tarkastella erillisellä, ohjelman mukana tulevalla analysointiohjelmalla. Ohjelma koostaa kerätyistä tiedoista tärkeimmät tunnusluvut, jotka se ilmoittaa maksimina, miniminä ja keskiarvoina eli laajoina numeerisina raportteina. Numeerisista tuloksista saa myös graafisen taulukon helpottamaan tulkintaa. Eri simulointiajojen samanaikainen vertailu ja tarkempaa analyysiä varten Excel-tiedostoihin talletetut tulokset mahdollistavat kattavan analyysin teon. (Promodel 2003)

5.2.2 Access

Microsoftin Access on relaatiotietokantaohjelma, jota voidaan käyttää itsenäisesti ja erillään erilaisten rekistereiden ja tietokantojen ylläpitoon sekä erilaisiin sovelluksiin. Access sisältää erilaisia työkaluja ja ohjaimia, joilla voidaan liittää esimerkiksi Excel-, Lotus-, Xbase- ja Paradox-tiedostoja Accessin omiin tietokantatauluihin. SQL-sovellukset ja ohjelmat voivat myös käyttää Access-tietokantoja hyväkseen. Access antaa mahdollisuuden hyödyntää VBA-ohjelmointikieltä, jolla voidaan mukauttaa sovellusta omiin tarpeisiin. Accessistä löytyy myös ohjattuja toimintoja erilaisten taulujen ja raporttien luomiseen. Uusin versio on MS Access 2002, jossa on parannettuja ominaisuuksia myös suoraan WWW-sivuille liitettävään käyttöön sekä uutuuksina mahdollisuutena käyttää puheentunnistusta käyttöliittymänä. (Microsoft 2002)

5.3 Rajapinta simulointiin

5.3.1 Active-X

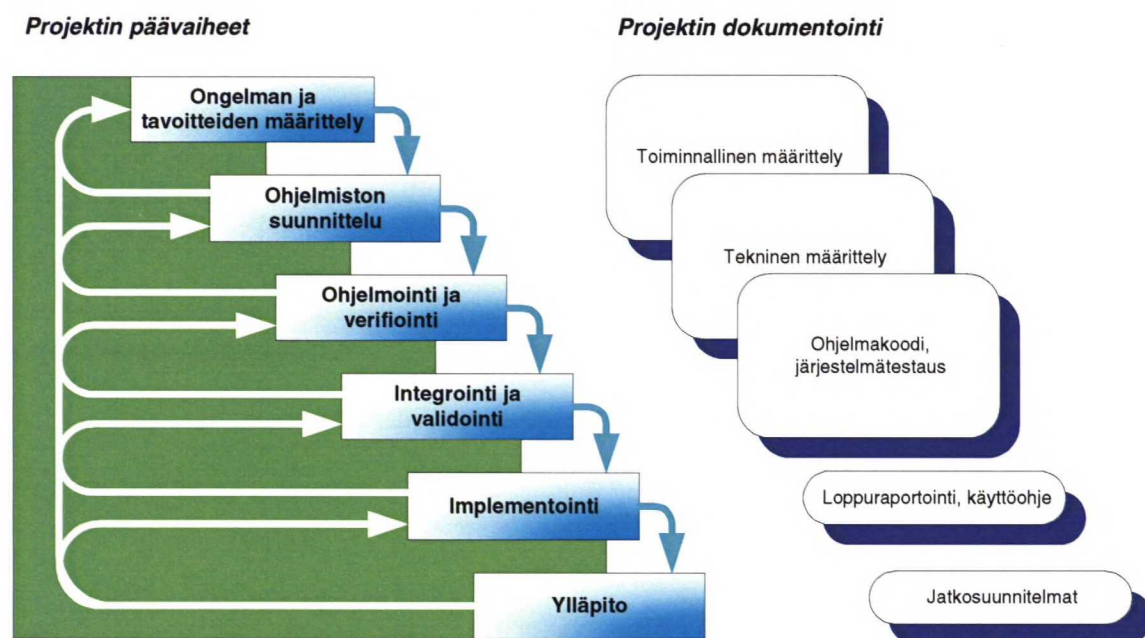
Active-X toimii ohjelmointikielenä eri sovellusten rajapinnoilla. Se yhdistää linkittämällä eri sovellusten tietoja toisiinsa Active-X:n kommunikoinnin avulla. Active-X-tekniikka on osaksi edeltäjiensä OLE- ja COM-tekniikoiden yhdistelmä. Itsenäiset Active-X-tuetut ohjelmat ovat prosesseiltaan erilaisia, joten niiden väliin tarvitaan yhteysprosessi eli protokolla, joka välittää kommunikointia sovellusten välillä. Rajapintana voidaan tapauksesta riippuen käyttää joko ohjelmien yhteistä linkitettyä objektikirjastoa, jota sovellukset kutsuvat, tai systeemikutsuilla yhteistä käyttöjärjestelmää hyväksikäyttäen. Active-X mahdollistaa myös koneiden väliseen kommunikointiyhteyden verkon kautta. (Chappell 1996, s.1-6)

6 PROJEKTIN TOTEUTUS

Tässä luvussa eritellään käyttöliittymä- ja tietokantaprojektin, eli yhteisellä nimellä kutsuttuna ohjelmistoprojektin eri vaiheet ja toteutus.

6.1 Projektin vaiheet

Käyttöliittymä- ja tietokantaprojektin systemaattinen työskentely voidaan jakaa luonteeltaan toisistaan eroaviin työvaiheisiin. Ghezzi, Jazayeri ja Mandrioli (1991, s.372) jakavat ohjelmistoprojektin kuuteen, kuvassa 13 esitettyyn päävaiheeseen. Päävaiheet ovat mukaelmia Boehmin (1988) kehittämästä vesiputousmallista.



Kuva 13. Projektin vesiputousmallin mukaiset vaiheet Ghezzi et al (1991, s.372) mukaillen.

Käytännössä projekti ei koskaan etene suoraviivaisesti vaiheesta toiseen edellisen työvaiheen päätyttyä. Kuvan 13 työvaiheet käsitelläänkin suurelta osin rinnakkain, jolloin työn eri vaiheiden valmistumiset voivat vaihdella suurestikin projektista toiseen. Esimerkiksi ohjelmoinnin työmäärän jakaminen monelle ihmiselle vähentää ohjelmointiin ja verifiointiin käytettyä aikaa, mutta vastaavasti lisää työmäärää integroinnissa ja validoinnissa. Rinnakkain käsiteltynä integrointia voidaan tehdä sitä mukaa kun eri ohjelmointiosat valmistuvat. Lisäksi projekti etenee iteratiivisesti, eli projektin edetessä voidaan usein joutua palaamaan edellisiin vaiheisiin. Esimerkiksi oh-

jelmoitaessa saatetaan todeta sovelluksen määrittelyt puutteellisiksi, jolloin niitä joudutaan tarkentamaan. (Koivunen 1994, s.19-22) (Royce 1998, s.121-123)

Teoriassa, vesiputousmallin mukaan, projektin dokumentointi on työvaiheeseen sidottu, ja itse työvaiheen lopputulos on siitä syntynyt dokumentti. Näin ollen seuraava työvaihe aloitetaan edellisestä vaiheesta syntyneen dokumentin pohjalta. (Ghezzi et al 1991, s.368) Usein käytännössä dokumentointi kulkee kuvan 13 mukaisesti rinnakkain eri työvaiheiden kanssa sekä osittain myös päällekkäin, kuten silloin kun toiminnallista määrittelyä muutetaan tekniseksi määrittelyksi arkkitehtuurisuunnittelun avulla.

Kappaleissa 6.2 - 6.7 kerrotaan tarkemmin projektin päävaiheista.

6.2 Ongelman ja tavoitteiden määrittely

6.2.1 Ongelman määrittely

Onnistunut projekti lähtee aina ongelman, tavoitteiden ja rajausten määrittelystä.

Ongelman määrittely on koko projektin kannalta tärkein vaihe. Keskeisin prosessin tuloksena syntyvä dokumentti on toiminnallinen määrittely, johon dokumentoidaan sekä vaatimukset että vaatimukset täyttävän järjestelmän kuvaus. Toiminnallisessa määrittelyssä kuvataan ohjelmiston toiminnot, toteutukselle asetettavat ei-toiminnalliset vaatimukset sekä rajoitukset. Ei-toiminnallisia vaatimuksia ovat esimerkiksi suoritusteho ja käytettävyys. Rajoituksina voivat olla puolestaan käytettävissä oleva järjestelmä tai toteutus tietyllä ohjelmointikielellä. Määrittelyvaiheessa on siis kysymys asiakasvaatimusten muuttamisesta täsmällisiksi ohjelmistovaatimuksiksi. Ongelman määrittely tulee muotoilla niin, että kaikki osallistuvat osapuolet hyväksyvät ongelman ja ovat yhtä mieltä sen määrittelystä. (Haikala & Märijärvi 1998, s.27, s.64-65)

6.2.2 Tavoitteiden asettaminen

Kun ongelma on määritelty, voidaan seuraavaksi määritellä projektille asetettavat tavoitteet. Tavoitteiden on oltava konkreettisia suureita, joita voidaan mitata täsmällisesti, kuten esimerkiksi: (Pelin 1996, s.77)

- aikatavoite (välitavoitteet)
- kustannustavoite
- tekniset tavoitteet ja laatu

Tavoitteiden mitattavuus on perusedellytys projektin onnistumisen analysointiin ja laadun valvontaan.

6.2.3 Projektisuunnitelma

Projektin alussa laaditaan projektisuunnitelma, joka kertoo, miten projektille asetetut tavoitteet on tarkoitus saavuttaa: mitä tehdään, kuka tekee, milloin ja miten. Projektin valvonta perustuu projektisuunnitelmaan. Projektisuunnitelma sisältää muun muassa: (Pelin 1996, s.75-76)

- projektin tavoitteet, rajaukset ja mahdolliset liittymät ympäristökuvauksiin
- projektiryhmän ja yhteyshenkilöt
- projektin työvaiheet ja aikataulusuunnitelman
- projektin resurssisuunnitelman ja budjetin

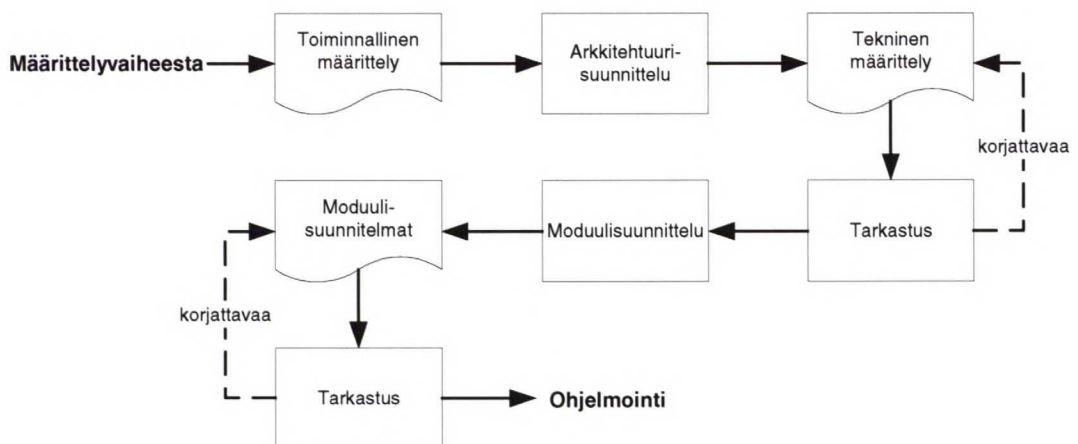
Projektisuunnitelma voi muuttua myös työn edetessä. Esimerkiksi työmäärää ja aikataulua voi olla hankala arvioida etukäteen työtä aloittaessa. Projektisuunnitelmaa joudutaan tarkentamaan aina tarpeen tullen. Sen tulisi kuitenkin olla realistinen ja vastata laatimishetkellä parasta tietämystä.

6.3 Ohjelmiston suunnittelu

Suunnitteluvaiheessa suunnitellaan määrittelyn kuvaamien toimintojen toteutus. Usein määrittelyvaiheessa kuvataan, *mitä* järjestelmä tekee, kun taas suunnitteluvaiheessa vastaavasti kuvataan *miten* se tehtävänsä suorittaa. (Haikala & Märijärvi, 1998 s.28-29)

Suunnitteluvaihe lähtee konseptisuunnittelusta, jossa järjestelmän toiminta muotoillaan luonnollisella ja ymmärrettävällä käsitteistöllä toiminnalliseksi kuvaukseksi. Konseptisuunnittelu toteutetaan työkaluriippumattomilla kuvaustekniikoilla, joiden avulla kuvausta laativat henkilöt voivat kommunikoida keskenään ja analysoida syntyntä suunnitelmaa. Luodut konseptisuunnitelmat voivat olla työdokumentteja seuraavalle vaiheelle tai jäädä lopulliseen dokumentaatioon kuvaamaan järjestelmää. (Hix & Hartson 1993, s.132 –133) (Haikala & Märijärvi, 1998 s.53)

Yleisen konseptisuunnittelun jälkeen suunnittelu jaetaan usein kahteen tasoon: arkkitehtuurisuunnitteluun ja moduulisuunnitteluun. Arkkitehtuurisuunnittelussa järjestelmä jaetaan mahdollisimman vähän toisistaan riippuviin moduuleihin, ja moduulien rajapinnat määritellään. Moduulisuunnittelussa suunnitellaan kunkin moduulin sisäinen rakenne, ja lopuksi eri moduulit liitetään yhteen määriteltyjen rajapintojen avulla. Arkkitehtuurisuunnittelun keskeisin tulos on siitä syntynyt ohjelmiston tekninen määrittely. (Haikala & Märijärvi 1998, s. 67-69) Kuvassa 14 on esitetty suunnittelu-prosessi konseptisuunnittelun kaaviona.



Kuva 14. Suunnittelu-prosessi Haikalaa & Märijärveä (1998, s.68) mukaillen.

6.4 Ohjelmointi ja verifiointi

6.4.1 Ohjelmointi

Kun järjestelmän rakenne on huolellisesti suunniteltu, voidaan aloittaa sen ohjelmointi. Suunnittelussa muodostettuja eri moduuleita ohjelmoidaan toisistaan erillisinä niin, etteivät yksittäiseen moduuliin tehdyt muutokset vaikuta muiden moduulien toimintaan. Ohjelmointi voidaan toteuttaa yleisempiä ohjelmointikieliä tai valmiita ohjelmistotyökaluja käyttäen.

6.4.2 Verifiointi

Verifiointi eli teknisen määrittelyn ja ohjelmoinnin vastaavuuden tarkastaminen on jatkuva prosessi ohjelmoinnin kehityksen edetessä. Verifiointi tarkoittaa siis koodin oikeellisuuden vertaamista sitä vastaavaan määrittelyyn. Oikeellisuuden tarkastus tapahtuu ohjelman testauksella aina pienistä osamoduuleista koko ohjelman läpi käymiseen. Verifiointin suorittaminen on käytännössä liian myöhäistä, jos se suoritetaan vasta ohjelman valmistuttua. Mahdolliset virheet ovat paljon vaikeampia ja kalliimpia korjata myöhäisessä vaiheessa kuin ohjelmoinnin alkuvaiheessa. (Ghezzi et al 1991, s.256)

6.5 Integrointi ja validointi

6.5.1 Integrointi

Integroinnissa ohjelman eri moduulit yhdistetään yhdeksi osaksi eli lopulliseksi tuotteeksi. Yhdistämiseen on jokaisessa osamoduulissa määritelty rajapinta, jonka mukaan integrointi voidaan toteuttaa. Integrointi voidaan suorittaa koko ohjelmalle yhdellä kertaa tai paloittain, moduuli kerrallaan. Paloittaisella integroinnilla on etuja verrattuna koko ohjelman samanaikaisesti integrointiin: (Ghezzi et al 1991, s.292)

- Integrointia voidaan toteuttaa heti moduuleiden tarpeellisen määrän valmistuttua, vaikka kaikki moduulit eivät olisi vielä valmiina
- Virheiden löytäminen on systemaattisempaa, kun jo integroituihin ja virheettömiin moduuleihin lisätään uusia moduuleita.

- Osasta moduuleista voidaan saada jo integroitua tarkoituksenmukainen toimiva osa, joka voidaan toimittaa osatoimituksena asiakkaalle etukäteen.

6.5.2 Validointi

Validointi on toimenpide, jolla pyritään osoittamaan, että toteutettava järjestelmä vastaa asiakkaan tarpeita. Projektin loppuvaiheessa vaatimusten toteutuminen voidaan varmistaa testaamalla tuotetta sen oikeassa käyttöympäristössä ja todeta se kuvaavan todellisuutta riittävän tarkalla tasolla. Validointia voidaan tehdä myös prosessin aikaisemmissa vaiheissa esimerkiksi prototyyppien avulla. (Haikala & Märijärvi 1998, s.84)

6.6 Implementointi

Ennen ohjelman lopullista valmistumista käydään läpi testausvaiheet, joista kerätään testitulokset. Onnistuneiden testitulosten perusteella voidaan suorittaa itse käyttöönotto eli implementointi. Ohjelman implementointi sisältää itse käyttöönoton lisäksi käyttöohjeen laatimisen ja mahdolliset opetus- tai esittelytilaisuudet. Nämä auttavat ohjelman käyttöönottoa. (Ghezzi et al 1991, s.367)

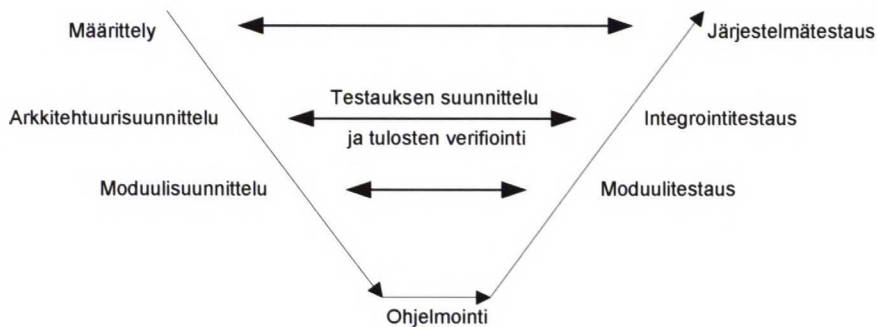
6.7 Ylläpito

Ylläpito on asiakkaan ongelmien ratkomista, virheiden korjaamista, ohjelman muuttamista vaatimusten muuttuessa sekä uusien piirteiden lisäämistä. Käytännössä ylläpito on kustannuksiltaan yli 60 prosenttia alkuperäisen ohjelman hinnasta. Siitä noin 20 prosenttia on virheenkorjausta sekä ympäristön muutoksiin mukauttamista ja yli 50 prosenttia uusien piirteiden lisäämistä. Valmiin ohjelman jatkokehitys on hyvä määritellä annettavaksi ylläpidon ulkopuolelle. Myöhemmin ohjelmaan ehdotetut muutokset ja lisäykset toteutetaan ohjelman seuraavassa versiossa, joka toteutetaan uutena projektina. (Haikala & Märijärvi 1998, s.29) (Ghezzi et al 1991, s.367, s.373)

6.8 Testaus

Testaus on monitasoinen toimenpide, jolla pyritään löytämään mahdolliset virheet ennen lopullista käyttöönottoa. Jokainen käyttöönoton jälkeen huomattu virhe on kal-

liimpaa korjata kuin itse testausvaiheessa huomattu. Virheiden voidaan ajatella jakautuvan vähintäänkin kahteen kategoriaan: vikoihin ja häiriöihin. *Vialla* tarkoitetaan toimintoa, joka voi ilmaantuessaan aiheuttaa järjestelmälle ongelmia. Usein se on harmiton ja voi korjautua itsessään tai toisen toiminnon kautta, eikä välttämättä johda häiriöön. Vain pahimmassa tapauksessa vika aiheuttaa näkyvän *häiriön* ulkoiselle toiminnalle, jolloin järjestelmän käyttö voi keskeytyä. Testauksessa on syytä pyrkiä järjestelmällisyyteen eri tasojen testaamisessa, alkaen ohjelmakoodin moduulitestauksesta aina koko järjestelmän testaukseen. Kuvassa 15 on esitetty V-mallin mukaiset testaustasot, joissa testauksen suunnittelu tapahtuu testaustasoa vastaavalla suunnittelutasolla. (Haikala & Märijärvi 1998, s.262-263)



Kuva 15. V-mallin mukaiset testaustasot.

Yksittäiset moduulitestaukset toteutetaan vertaamalla moduulin toimintaa tekniseen määrittelydokumenttiin ja testien suorittajana on usein moduulin toteuttaja itse. Yksittäiseen moduuliin kuluu noin 100 – 1000 ohjelmariviä, ja sitä voidaan testata erilaisin ajurein tai testiohjelmin.

Integroititestauksessa tarkastellaan puolestaan moduuleista koostuvia ryhmiä, joiden kohdalla painopiste on moduuleiden välisten rajapintojen toimivuuden testaamisessa. Tässä verrataan moduuliryhmien toteutusta arkkitehtuurisuunnittelun pohjalta luotuihin dokumentteihin. Integroititestausta etenee usein moduulitestauksen rinnalla, ja sitä onkin yleensä tarpeetonta tarkastella erillään moduulitestauksesta. Integroititestausta etenee tavallisesti kokoavasti eli alimman tason moduuleista ylöspäin.

Järjestelmätestauksessa taas koko järjestelmän toimintaa verrataan määrittelydokumentaatioon ja asiakasdokumentaatioon. Testaajiksi on syytä valita kehitystyöstä erillään olevia, mahdollisimman riippumattomia testaajia. Järjestelmätestauksessa

testataan myös ei-toiminnallisia ominaisuuksia, kuten kuormitusta, luotettavuutta ja käytettävyyttä. (Haikala & Märijärvi 1998, s.263-264)

6.9 Projektin aikataulun ja työmäärän arviointi

Työmäärän arviointi on suunniteltavan aikataulun perusta. Aikataulut arvioidaan aina kunkin hetken parhaan tietämyksen ja kokemuksen mukaisesti. Uusien projektien perusteellisemman aikataulun tueksi voidaan käyttää aikaisemmista vastaavista projekteista kerättyjä tuntitilastoja. Projektien jako työmäärän mukaisiin välivaiheisiin helpottaa myös aikataulun seurantaa sekä jaksotusta. Projektin aikataulun virheellinen, usein liian tiukka määrittely ja tämän seurauksena kasvavat riskit kostahtavat tuotteen elinkaaren myöhemmissä vaiheissa moninkertaisesti. (Pelin 1996, s142-148)

6.10 Projektin arviointi

Projektin päätyttyä tehdään projektista arviointi, jossa käydään läpi projektin aikana tulleet onnistumiset ja epäonnistumiset sekä tavoitteiden saavutus. Samalla selvitetään syyt epäonnistumisiin ja kehitetään tämän pohjalta parempia menetelmiä. Myös kokemuksen kasvaminen projektin myötä antaa työntekijöille lisäarvoa, jota voidaan käyttää myöhemmin hyväksi. Projektin arvioinnissa on mahdollista käyttää eri laatu-tavoitteiden saavuttamisen tutkimiseen tarkoitettuja keinoja kuten esimerkiksi heuristista laatuarviointia tai sertifioituja laatustandardeja, joiden avulla työtä voidaan kehittää oikeaan suuntaan. (Pelin 1996, 43-54)

7 OLEMASSA OLEVIENT JÄRJESTELMIEN TUTKIMINEN JA VERTAILU

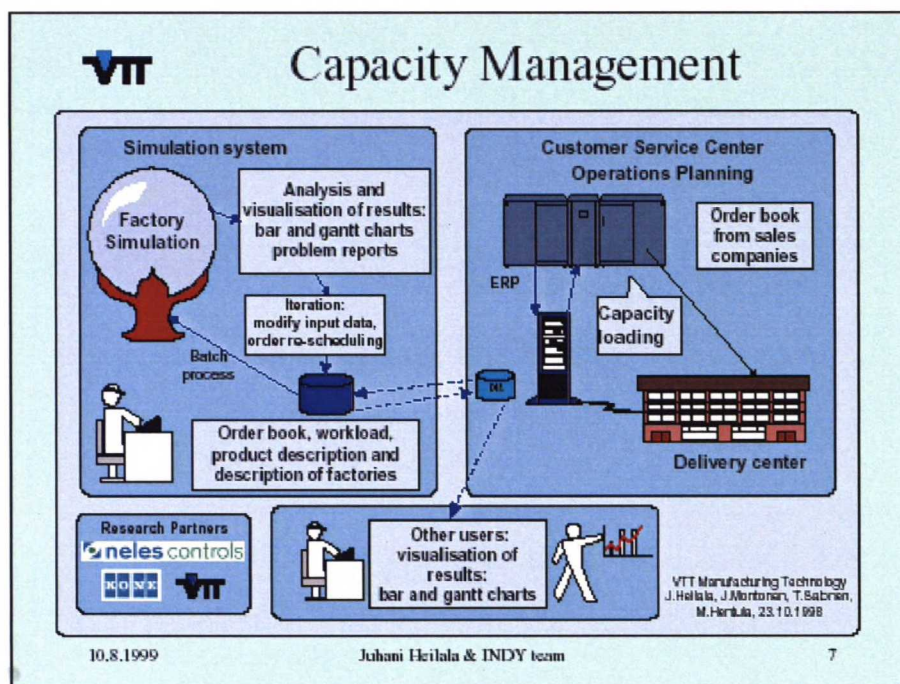
Erilaisia, vertailtavia versioita simuloinnin käyttöliittymäjärjestelmistä on jo olemassa. Tulevaisuudessa, simuloinnin tunnettavuuden ja käytön lisääntyessä, simuloinnin tarve jokapäiväisessä käytössä kuitenkin kasvaa yhä voimakkaammin. Ohjelmien kehittyessä simuloinnin käyttö operatiivisessa toiminnassa ja mitoituksessa lisää suunnitelmallisuutta tulevaisuuden ennustamisessa. Lisääntynyt arkipäiväinen käyttö edellyttää siten simuloinnilta sen käyttöliittymän ja tietokantojen yksinkertaista ja nopeaa hallittavuutta.

Olemassa olevien järjestelmien tutkimiseen valittiin kaksi järjestelmää: Indy- ja IndyNet-järjestelmät, jotka ovat VTT:n kehittämiä pilottiprojekteja. Valittujen järjestelmien valintakriteereinä oli diplomityön käytännönsan mukainen, asiakaskohtainen räätälöinnin mahdollisuus. Tällöin monet muut valmiit järjestelmätuotteet eivät tulleet kyseeseen. Lisäksi valitut järjestelmät on mahdollista liittää suoraan osaksi erilaisiin toiminnanohjausjärjestelmiin.

Kummatkin esiteltävät järjestelmät on rakennettu hyödyntämään omaa tietokantaa ja graafista käyttöliittymää operatiivisen ohjauksen tukena. Valitut järjestelmät kuvaavat kahta peräkkäistä projektia, joista ensimmäinen projekti, Indy tehtiin vuosina 1997-1999 ja toinen projekti, IndyNet kehitettiin vuosina 2000-2001. Kappaleissa 7.1 - 7.2 kerrotaan tarkemmin VTT:n erikoistutkija Juhani Heilalan haastatteluun pohjautuen kyseisistä projekteista, joissa hän toimi projektipäällikkönä.

7.1 Indy-projekti

Operatiivisen toiminnan hallintaan rakennettu integroitu dynaaminen verkostosimulointimalli eli Indy-projekti oli osittain Tekesin rahoittama VTT:n tutkimus, jossa yhteistyössä mukana olivat Neles Controls Oy ja Kone Oyj. Kuvassa 16 on esitetty projektin toimintamalli.



Kuva 16. Indy-projektin toimintamalli (Heilala et al 1999).

Indy-projekti kehitti työkalun toimitusketjun muuttuvan kapasiteettitarpeen hallintaan. Työkalua voidaan hyödyntää asiakasohjautuvan tuotannon suunnittelussa. Sitä käytetään komponenttitehtaiden ja tuotantolinjojen karkeakuormituksen suunnitteluun, jossa ajoitusta muutetaan graafisella käyttöliittymällä. Järjestelmän avulla komponenttien valmistusta voidaan synkronoida, jolloin toimitustäsmällisyys paranee. Järjestelmä raportoi käyttäjälle potentiaalisista tulevista ongelmista, kuten esimerkiksi osoittaa myöhästyvät toimitukset etukäteen tai antaa tiedon tuotantoresursien ylikuormitustilanteesta. Järjestelmällä saadaan näkymä tulevaisuuteen, mikä mahdollistaen mm. uusien asiakasprojektien oikean ajoituksen jo tarjousvaiheessa. (Heilala et al 1999)

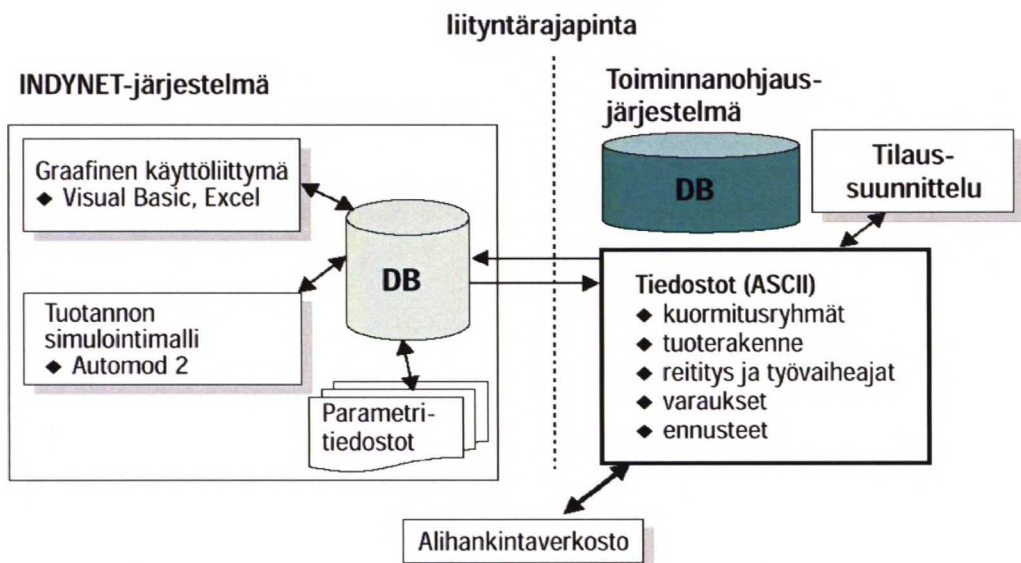
Työkalu mahdollistaa muutokset ilman simulointi-insinööriä ja usein myös yhden henkilön tietotaitoa operatiivisesta kapasiteettihallinnasta. Työkalu on myös integroitavissa osaksi yrityksen toiminnanohjausjärjestelmää. Työkalu on asennettu yhteistyössä olleiden yritysten käyttöön vuosina 1998-1999. (Heilala 2002)

Indy-järjestelmä perustuu tapahtumapohjaisen simuloinnin ja perinteinen tuotannonohjauksen integroimiseen. Se toimii PC-tietokoneessa Windows-ympäristössä. Simulaattori ja muut järjestelmän osat siirtävät tietoa ASCII-tekstitiedoston avulla. Käyttöliittymä on rakennettu Visual Basic:llä ja osittain myös

Käyttöliittymä on rakennettu Visual Basic:llä ja osittain myös Excel-välilehtien avulla. (Heilala 2002)

7.2 IndyNet-järjestelmä

IndyNet-projektin lyhenne merkitsee integroitua dynaamista asiakasohjautuvaa tuotantoverkoston hallintaa operatiivisen simuloinnin avulla. Lyhyesti sanottuna IndyNet kehittää prosesseja ja ohjelmistoja tavoitteenaan alihankintaverkoston toimitusprojektien hallinta ja tehostaminen. Projekti toteutettiin raskaan konepajateollisuuden verkostotuotannon suunnitteluun, analysointiin sekä ohjaukseen nopeasti muuttuvissa olosuhteissa, joissa toimitusaika on tulevaisuudessa merkittävä kilpailutekijä. Kuvassa 17 on esitetty yksi IndyNet-järjestelmän malli, joka kehitettiin Wärtsilälle. (Heilala 2002)



Kuva 17. IndyNet-järjestelmän rakenne Wärtsilälle (Heilala 2002).

Kuvassa projektin simulointimallille voidaan syöttää lähtötietoja parametreina, tietäen rajoituksin. Tällaisia parametreja voivat olla esimerkiksi resurssit, reititykset ja työvaiheet. Graafisen käyttöliittymän ja taulukkolaskennan avulla annetut parametrit vietään simulointimalliin, joka antaa tuloksia analyysiä varten. Graafinen käyttöliittymä ei edellytä ohjelmointi- tai simulointiasiantuntijaa. Tulokset saadaan vietyä IndyNet-järjestelmästä toiminnanohjauksen puolelle tuomalla ne välitiedostoon

ASCII muodossa. Sieltä tieto saadaan kerättyä päätöksenteon tueksi esimerkiksi Excel laskentaa hyväksikäyttäen. Hyväksytyt tulokset on myöhemmin mahdollista päivittää toiminnan ohjausjärjestelmään. (Heilala 2002)

Edellä esitetyt projektit ovat keskenään tekniikoiltaan ja toimintaperiaatteiltaan samanlaisia. Tavoitteeltaan IndyNet on viety pitemmälle verkostotasolle Indyprojektiin verrattuna. Käytetyt sovellukset ovat lähes samoja, mutta niissä on pientä vaihtelua asiakkaittain toivomusten mukaan. Työkaluina on käytetty Windows-ympäristön toimisto-ohjelman sovelluksia kuten Excel-taulukkolaskentaa. Tällaiset ohjelmat ovat yleisiä ja edullisia ratkaisuja projekteissa. Simulointiohjelmasta annetaan loppukäyttäjälle vain run-time versio, jolloin ohjelman käytön lisenssimaksut ei muodostu merkittäväksi tekijäksi projektin kustannuksille. Graafinen käyttöliittymä on rakennettu Visual Basicilla, joka kestää paremmin ohjelmien päivityksen tulevaisuudessa.

8 HISSISIMULOINNIN TYÖKALUSOVELLUS

Diplomityön soveltavassa osassa rakennettiin henkilöhissijärjestelmän mitoittamiseen työkalusovellus käyttäen hyväksi tapahtumapohjaista simulointia. Tässä luvussa kerrotaan työkalusovelluksen taustasta, tavoitteista ja projektin toteuttamisesta käytännössä. Luvussa 9 esitellään lähemmin diplomityön tuloksena syntyneen työkalusovelluksen käyttöönottoa osana kauppakeskus Jumbon sisälogistiikkaa. Luvussa 10 arvioidaan koko projektin onnistumista ja saavutettuja hyötyjä.

8.1 Työkalusovelluksen tausta, tarve ja tavoitteet

EP-Logistics Oy on osallistunut usean kauppakeskuksen henkilöliikenteen suunnitteluun, jolloin yrityksellä on ollut tarve hissiliikenteen mitoittustyökalulle.

Hissivalmistajien mitoitusohjeet ja Rakennustietosäätiön RT-ohjekortit on soveltuudeltaan tarkoitettu lähinnä toimisto-, virasto ja liikerakennusten, koulujen, hotellien, sairaaloiden sekä asuinkerrostalojen suunnitteluun. Edellä esitetyn kaltaiseksi rakennuksiksi voidaan luokitella kaikki ne, joissa henkilöliikenteen katsotaan lähtevän tietystä kerroksesta ja jakaantuvat eri kerroksiin, tai takaisinpäin kootusti: liikennettä on eri kerroksista samaan lähtökerrokseen. Tällaisesta on hyvänä esimerkkinä toimistorakennus: aamulla kaikki tulevat samasta lähtökerroksesta sisään, mutta menevät töihin eri kerroksiin. Työpäivän jälkeen eri kerroksista lähdetään saman tulo-kerroksen kautta pois.

Ostoskeskusten hissimitoitus on hieman erilainen verrattuna toimistorakennusten mitoittamiseen. Ostoskeskuksessa lähtökerroksia on yleisesti useita, koska esimerkiksi paikoitustilat, joista henkilöt saapuvat rakennukseen sijaitsevat eri kerroksissa. Samoin henkilöt jakaantuvat eri kerroksiin myös pois lähdettäessä. Toinen ongelma RT-ohjekortin käytölle ostoskeskuksissa on kerroskorkeuden normaalia suuremmat erot. RT-ohjekortin kaavioita voidaan käyttää vain silloin, kun kerroskorkeudet ovat 3,0 - 3,6 metriä jokaisessa kerroksessa. Isoissa ostoskeskuksissa kerroskorkeudet ylittävät usein nämä RT-ohjekortissa vaaditut mitat. Tällaiseen mitoittamiseen simulointi on työkalusovelluksena selkeästi tarkempi analysoinnin työväline, kuin turvau-

tuminen kokemukseen pohjautuvaan arvioon tai vertailulaskelmiin erilaisista ostoskeskuksista.

Toteutetun diplomityön työkalusovelluksen tavoitteena on, EP-Logistics Oy:n simuloinnin kehitystyönä, vastata juuri kauppakeskusten hissiliikennesuunnitteluun liittyviin tarpeisiin. Diplomityön tavoite on toimia myös käytettävissä olevana pilotti-projektimallina ja toteutuksen ohjeellisena dokumentaationa muiden simulointimallien työkalusoveltamiseen tulevaisuudessa.

Kappaleissa 8.2 - 8.3 kerrotaan ensin simuloinnin käytännön toteutuksesta ja seuraavissa kappaleissa 8.4 – 8.6 simuloinnin integroinnista työkalusovellukseksi.

8.2 Simulointimallin määrittely ja mittarit

Tutkittavasta hissijärjestelmästä laadittiin hissien teknisten tietojen pohjalta konseptimalli yhteistyössä järjestelmän asiantuntijoiden kanssa. Simuloinnin avulla selvitettiin tarvittavaa hissien määrää ja kokoa osana henkilöliikenteen tasonvaihtojärjestelmää. Simulointimalliin määriteltiin kuuluvaksi seuraavat pääosat:

- tekniset parametrit
- henkilöliikennevirrat
- ohjausperiaatteet
- rajoitukset ja oletukset
- mittarit

Seuraavissa kappaleissa 8.2.1 - 8.2.5 määritellään tarkemmin simulointimallin pääosat ja niiden käyttö sekä malliin rakennetut mittarit.

8.2.1 Tekniset parametrit

Hissien toimintaan kuuluvat tekniset tiedot on määritelty taulukossa 3. Tietojen arvot on pystyttävä antamaan erilaisina kokeiltavina parametreinä simulointimallille. Tässä järjestelmässä jokaisella hissillä on käytössä samat annettavat parametrit. Hissien

henkilömäärä voidaan jättää myös rajoittamatta, jolloin yksittäisten hissien tilakoon ylittävää käyttöä voidaan seurata mittareilla.

Taulukko 3. Hissien tekniset ominaisuudet annettavina parametreinä.

parametri	yksikköä
hissien lukumäärä:	kpl
henkilölukumäärä:	hlö
ostoskärrylukumäärä:	kärryä
nimellisoikeus:	m/s
kiihtyvyys:	m/s ²
hidastavuus:	m/s ²
pysähdysaika:	s
käynnistysviive:	s
lukitsemisviive:	s

Lisäksi kauppakeskuksen ominaisuuksista parametrein muutettavia ovat kerroskorkeudet peräkkäisten kerrosten välillä. Kerrosten korkeudet ovat jokaiselle hissiryhmään kuuluvalla hissillä samat.

8.2.2 Henkilöliikennevirrat

Simulointimallille voidaan antaa lähtötietoina ihmisten liikkuminen tasojen vaihdossa kerroksittain. Tällaisia tietoja ovat mm. ostoskeskukseen tulevien ja sieltä lähtevien ihmisten määrät eri parkkikerroksissa ja liiketasoilla, sekä kerrostenvälinen, sisäinen tasonvaihtoliikenne. Tässä henkilöliikennettä tutkitaan tunnin mittaisina ajoina, jolloin voidaan tutkia esimerkiksi eri ruuhkatuntien vaikutusta hissiliikenteeseen. Kaikki hissiliikenne on mallinnettu henkilön ja ostoskärryn tarkkuudella.

8.2.3 Ohjausperiaatteet

Simuloinnissa hissien ohjaus tapahtuu ryhmäohjausjärjestelmän mukaisesti käyttäen täyskoontaohjausta. Tällöin hisseillä on kerroskohtaisesti kaksi yhteistä kutsupainonappia, joilla matkustaja voi ilmoittaa haluamansa ajosuunnan. Hissi noudattaa vain senhetkisen liikkeen mukaan suuntautuneita kerroskutsuja.

Hissin vapautuessa kaikista korin sisäpuolisista kerrostilauksista se vastaa kauimmin odottaneen hissitilauksen mukaiseen kerroskutsuun. Hissi jättää henkilöt heidän tekemiensä kerrostilausten mukaisiin kerroksiin. Simulointimallilla voidaan myös testata erilaisia priorisointeja ohjauksen tehostamiseksi.

8.2.4 Rajoitukset ja oletukset

Simulointimallilla on tässä projektissa käytössään seuraavia oletuksia ja rajoituksia:

- Ihmiset eivät vaikuta hissin pysähdysaikaan. Pysähdysaika on sama joka kerroksessa.
- Käytössä on enimmillään 10 kerrosta käytössä ja 4 hissiä yhdessä hissiryhmässä.
- Yksi ostoskärry, kärryn ohjaaja mukaan luettuna, vie 3 ihmisen tilan hississä.
- Hissien kapasiteetti on rajaton (niiden kokoa seurataan laskureilla) hissien kapasiteettitarvetta määriteltäessä. Hissien suunniteltu henkilökapasiteetti on 21 henkilöä/hissi.

8.2.5 Mittarit

Järjestelmän toiminnan seuraamiseksi malliin on rakennettu mm. seuraavia mittareita:

- Hissien käyttöasteet
- Hissien henkilömäärien koko
- Ihmisten jonotusajat hissille yhteensä
- Hissille jonottavien ihmisten määrä kerroksittain
- Hisseistä lähtevien ihmisten liukuva henkilömäärä tunnissa

Määrittelyn lisäksi projektin käynnistysvaiheessa laadittiin aikataulun, resurssien allokoinnin ja budjetin sisältävä projektisuunnitelma.

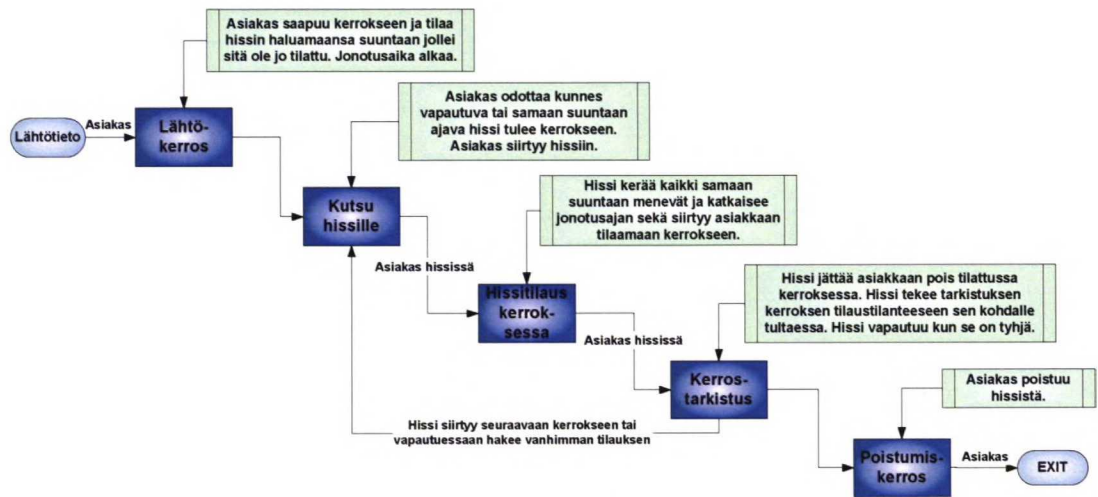
8.3 Simulointimallin suunnittelu ja toteutus

8.3.1 Simulointimallin lähtötiedot

Simulointimallin lähtötietoina käytettiin Excel-tilukkolaskennalle annettuja kerroskohtaisia määriä hissille saapuvista ihmisistä. Annettujen määrien perusteella muodostettiin ihmisten saapumisväli. Ihmisten saapumisvälin generoimiseen käytettiin eksponentiaali jakaumaa, joka kuvaa hyvin ihmisten mahdollista saapumista kerrokseen.

8.3.2 Simulointimalli

Laadittujen määritelmien pohjalta ohjelmoitiin hissijärjestelmän tietokonemalli. Tietokonemalli toteutettiin ProModel –simulointiohjelmiston versiolla 2001. Tietokonemallin verifiointi kuului kiinteänä osana mallin ohjelmoimintiin. Lopullisen, valmiin mallin validointi toteutettiin järjestelmän asiantuntijoiden kanssa. Mallin todettiin kuvaavan riittävän tarkasti todellista järjestelmää. Kuvassa 18 on esitetty hissinkinseptimallin rakenne pääpiirteittäin.

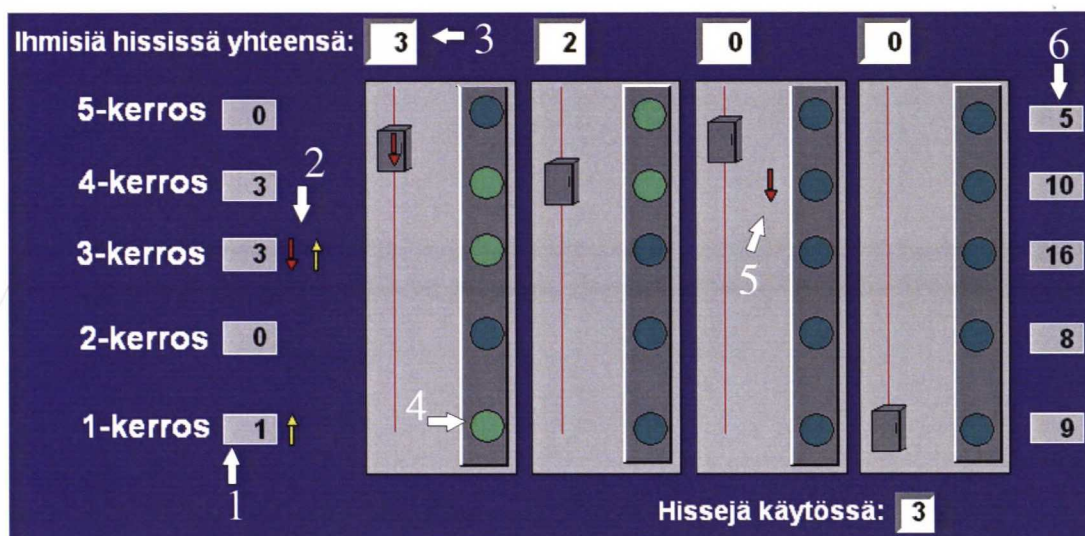


Kuva 18. Hissisimuloinnin konseptimalli.

Kuvassa 19 on esitetty simulointimallin graafinen kuvaus, jossa valkoisilla nuolilla on osoitettutärkeimmät kohdat:

- Kohdassa 1 mittari kertoo kerroksissa jonottavien ihmisten lukumäärän.

- Kohdassa 2 punainen nuoli alaspäin ja keltainen nuoli ylöspäin kertovat tehtyjen tilausten suunnan.
- Kohdassa 3 luku kertoo hississä olevien ihmisten määrän.
- Kohdassa 4 valot kertovat hissien sisällä annetut kerroskohtaiset korikäskyt eri kerroksiin. Vaalean vihreä palavaa valoa ilmoittaa kerrokset joihin ollaan vielä menossa.
- Kohdassa 5 tyhjä hissi kuittaa kyseisessä kerroksessa olevan tilauksen.
- Kohdassa 6 mittarit kertovat kerrokseen saapuvan liukuvan henkilömäärän/tunti.

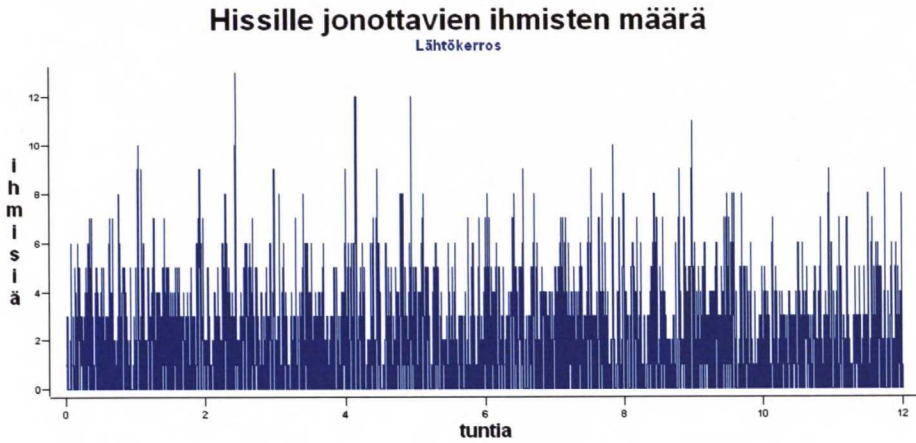


Kuva 19. Havaintoesitys simulointityökalulla ohjattavasta simulointimallista.

8.3.3 Simulointimallin tulokset

ProModel-simulointiohjelmisto sisältää tulosten analysointiin kattavan tulostilaston erilaisine kuvaajineen aikasarjoista, histogrammeista tai käyttöasteista. Tämä säästää aikaa tuloskuvaajien laskennalliselta toteuttamiselta, koska ohjelmiston kuvaajia voidaan hyödyntämään sellaisenaan. Hissisimuloinnissa tärkeimmät tuloskuvaajat on suoraan ohjelmoitu ilmestyväksi simulointiajon päätyttyä ymmärrettävinä kuvaajina, jolloin käyttäjä saa ne automaattisesti esille ilman ohjelman käytön osaamista. Simulointimallin tilastoista saa myös osittain itse luotua erilaisia kuvaajia käyttäen apuna

taulukkolaskentaa. Itse tehtyjä kuvaajia käytetään usein vertaillessa eri simulointiajoja keskenään. Kuvassa 20 on esimerkkikuvaaja ProModelin antamasta aikasarjasta, jota voi hyödyntää analyysin teossa. Kuvasta nähdään hissille jonottavien henkilöiden määrän yhdessä kerroksessa eri ajankohtina.



Kuva 20. Esimerkki ProModelin tulostuvaajasta.

8.4 Työkalusovelluksen määrittely ja tavoitteet

Työkalusovellus määriteltiin toteutettavaksi Windows-käyttöympäristöä tukevaksi käyttäen hyväksi yleisiä toimisto-ohjelmia. Nämä ohjelmat löytyvät useimmilta asiakailta ja oman toimiston henkilökunnalta ilman erillisiä ohjelmallisia investointeja. Tavoitteena oli integroida simulointiohjelma ja saadut tulokset yhteen käyttäjäystävälliseen käyttöliittymään sellaisille käyttäjille, joilla ei ole simuloinnin asiantuntemusta.

Työkalusovellukselle määriteltiin seuraavat toiminnalliset tarpeet:

- Työkalusovelluksen helppo asentaminen.
- Simuloinnille annettavien parametrien yksinkertainen muuttaminen ajokohdasta.
- Simulointiajojen käynnistys annetuilla parametreilla.
- Tulosten yksityiskohtainen esitys.

- Standardianalyysin luonti tehdystä simulointiajosta tulostettavaksi raportiksi.
- Analyysien tallennusmahdollisuus ajojen vertailuja varten.
- Käyttäjäystävällinen suomenkielinen käyttöliittymä.

Seuraavassa määritellään tarkemmin työkalusovelluksen tärkeimmät osat ja niiden toteutettu toiminta.

8.4.1 Työkalusovelluksen helppo asentaminen

Sovellukselta vaadittiin helppo asennus, jolloin yksinkertaisimmillaan se ohjautuu CD:ltä käyttäjän kovalevylle automaattisesti. Käyttäjälle tulee silti jättää mahdollisuus antaa sovellukselle itse asennusta koskevia määräyksiä, joiden mukaan asennus aloitetaan.

8.4.2 Parametrien antaminen

Parametrien syöttö jaettiin kolmeen osaan: henkilöliikenteen syöttöön, hissien tekniisiin ominaisuuksiin ja rakennuksen kerroskorkeuden määrittelyyn. Henkilöliikenne määritellään antamalla ihmisten lukumäärä ja kerros, mistä ihmisiä lähtee, sekä mihin kerrokseen he saapuvat.

Hissien teknisiksi parametreiksi määriteltiin sivulla 59 esitetyt taulukko 3:n kohdat, jotka rakennettiin jo hissisimulointia tehtäessä. Kerroskorkeuden määrittelyssä annetaan parametrein hissien kulkema matka eri kerrosten välillä. Kaikki parametrien syötöt tapahtuvat työkalusovelluksen eri näkymissä ja ohjautuvat sieltä automaattisesti simuloinnin lähtötiedoiksi tai simulointimallin parametreiksi.

8.4.3 Simuloinnin käynnistäminen ja ajojen suorittaminen

Simuloinnin käynnistys ja annettujen parametrien mukainen simulointiajo käynnistään automaattisesti työkaluohjelmasta annetulla käskyllä. Simulointiohjelman käynnistystä koskevat kohdat ohitetaan ohjelmallisesti niin, että käyttäjä ei niitä näe, eikä hänen tarvitse niistä huolehtia.

8.4.4 Tulosten esittäminen

Simulointiajon päätyttyä käyttäjä voi valita, haluaako tallettaa simulointiajon tulokset, ja haluaako katsella niitä. Tärkeimmät tuloskuvaajat on asetettu valmiiksi esille välilehdittäin, joita selaamalla voidaan tutkia tarkemmin eri aikasarjoja tai histogrammeja (ks. esimerkki kuvassa 20, s.63).

8.4.5 Standardianalyysin luonti ja tallennus ajojen vertailua varten

Simulointiajojen käyttäjäystävällistä tulkintaa varten määriteltiin standardianalyysi, joka kertoo yhdellä A4-arkilla tärkeimmät asiat simulointiajon tuloksista sekä ajossa käytetyt parametrit. Jokainen luotu standardianalyysi tallennetaan tietokantaan mahdollista myöhempää vertailua varten.

8.4.6 Käyttäjäystävällisyys

Käyttäjäystävällisyys määriteltiin teoriaosassa esitellyn heuristisen suunnittelun (ks. s.32) avulla, johtuen projektin tiukasta aikataulusta ja hyväksi havaitusta suunnittelukäytännöstä pienissä projekteissa.

Käyttäjäystävällisyydellä pyritään työkalusovelluksen nopeaan oppimiseen sekä sujuvaan ja vaivattomaan käyttöön. Lisäksi käytettävyydellä tehokkuudella ja virheettömyydellä sekä muistettavuudella pyritään antamaan käyttäjälleen mahdollisimman tyytyväinen käyttökokemus työkalusovelluksesta. Käyttäjiksi määriteltiin vain suomenkieliset ihmiset.

8.5 Sovelluksen suunnittelu ja toteutus

Edellä esitettyjen vaatimusten mukainen työkalusovellus toteutettiin käyttäen pääasiassa VBA-ohjelmointikieltä Access-tietokantaohjelmaan mukauttaen. Integroituna ProModel-simulointiohjelman ja työkalusovelluksen välillä toimivat Active-X:llä toteutetut parametrejä välittävät ohjelmakoodit sekä lähtötietoina henkilöliikenteen syötössä Excel-taulukkolaskenta. Seuraavissa kappaleissa esitellään tarkemmin työkalusovelluksen suunnittelua ja toteutusta.

8.5.1 Heuristinen suunnittelu

Suunnittelussa työkalusovelluksen ulkoasuun ja toimintaan kiinnitettiin huomiota teoriaosassa esitettyjen heurististen sääntöjen mukaisesti.

Yksinkertainen ja luonnollinen dialogi on saavutettu sijoittamalla näkyviin mahdollisimman vähän toimintapainikkeita. Ylimääräinen tieto, kuten asentamisen jälkeen ruudulla näkyvä ”Asenna”-painike, jota ei enää tarvita, on häivytetty muuttamalla painike harmaaksi, mikä tarkoittaa, että toiminto on poistunut käytöstä. Painikkeet on sijoitettu myös loogisesti käytön mukaiseen järjestykseen ylhäältä alas. ”Sulje ohjelma”-painike on erillään muista ja esitetty muista poiketen, jolloin sen tunnistaa helposti erilliseksi toiminnoksi. Kuvassa 21 on esitetty työkalusovelluksen aloitussivun ulkoasu.



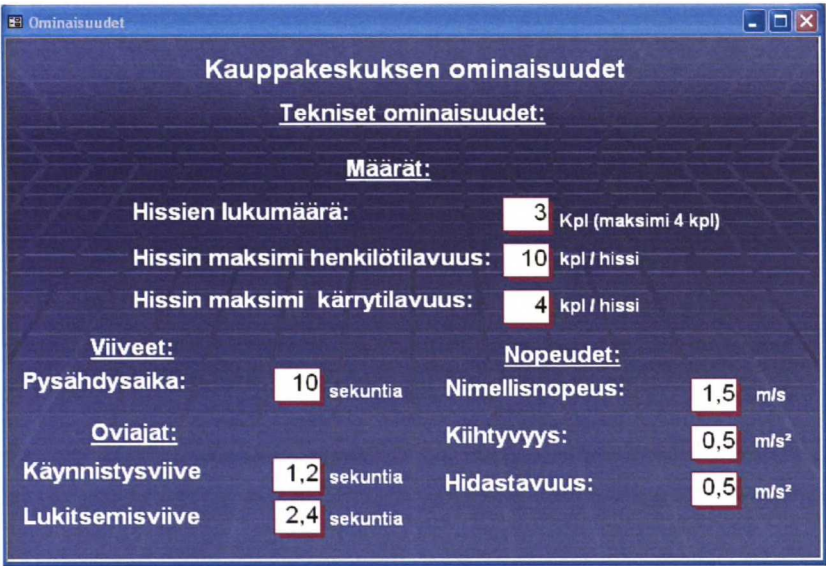
Kuva 21. Työkalusovelluksen pääsivun ulkoasu.

Käytetty kieli on termistöltään yksinkertaista, lähellä käyttäjän omaa kieltä, joten erilaiset ammattitermit on jätetty pois. Sovellus on suunniteltu määritelmän mukaan vain suomenkielisille käyttäjille.

Käyttäjän muistikuorman minimointi on otettu huomioon valikkojen määrän vähäisyydellä ja selkeällä asettelulla. Yhdestä painikkeen takaa aukeavasta valikosta ei päästä jatkamaan edelleen muihin valikkoihin, vaan sieltä palataan aina pääikkunaan

takaisin. Tämä yksinkertaistus auttaa käyttäjää muistamaan helposti minkä painikkeen takaa haluttu valikko löytyy ja mistä sinne päästään.

Käyttöliittymän osien sijoittelu on aina samantyyppinen ja yhdenmukainen eri näkymissä. Esimerkiksi oikeassa yläkulmassa on valikon sulkemiseen oma painike, joka on yhdenmukainen Windows-sovelluksissa käytetyt painikkeen kanssa. Samoin jokainen valikko sisältää otsikon ja asetettavat arvot jaoteltuina yhdenmukaisesti omiin alaotsikkotasoon ryhmiin ylhäältä alaspäin lukien kuten kuvassa 22 on esitetty.



Kuva 22. Yhtenäinen käytäntö valikkosivuilla.

Palautteen antaminen käyttäjälle parametrien asetuskomentoa suoritettaessa on viiveetön. Tällöin erillistä palautetta ei tarvita. Simulointi sen sijaan vie enemmän aikaa. Tällöin käyttäjälle näytetään visualisointi suoritettavasta simuloinnista; näin käyttäjä tuntee hallitsevansa järjestelmää. Simulointiajon seuraamisen sijasta on mahdollista odottaa simuloinnin päättymistä pääikkunassa, jolloin käyttäjälle ilmoitetaan erikseen simuloinnin olevan käynnissä.

Pääikkunassa oleva ”Sulje ohjelma”-painike on selkeä poistumistapa koko järjestelmästä. Muut normaalit poistumistavat ovat esillä ikkunan oikeassa yläkulmassa.

Vaihtoehtoiset oikopolut on huomioitu sovelluksessa parametrejä asetettaessa. Tällöin esimerkiksi kuvan 22 mukaisista numeroarvoista seuraavaan parametriin pääs-

tään suoraan sarkainnäppäintä painamalla. Näin hitaana toimenpiteenä tunnettu hiirellä kohdistaminen ohitetaan.

Käyttäjälle annetaan selkeä virheilmoitus, jos syötetty parametrin arvo sallittujen arvojen ulkopuolella. Esimerkiksi hissejä voidaan asettaa käyttöön vähintään 1 ja korkeintaan 4 kappaletta. Jos käyttäjä syöttää jonkin arvon välin ulkopuolelta, sovellus antaa virheilmoituksen, jossa ohjataan käyttäjää asettamaan oikeanlainen arvo. Näin käyttäjän mahdollinen väärä toiminta estetään jo ennen kuin se aiheuttaa virhetilanteen simulointiohjelmassa.

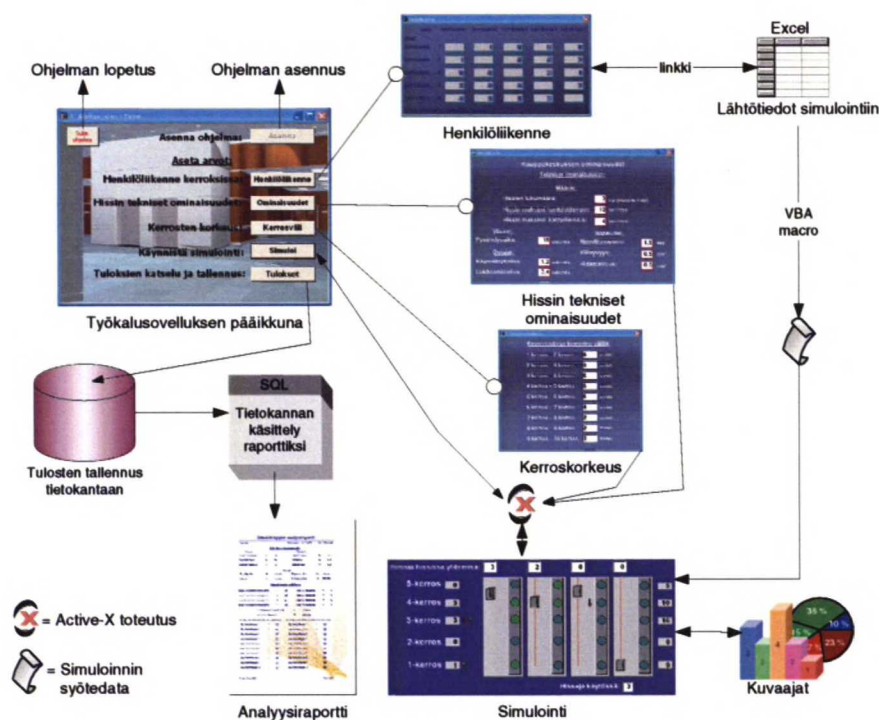
Käyttöliittymästä on tehty mahdollisimman selkeä ja yksinkertainen, jolloin sitä tulisi pystyä käyttämään ilman ohjeita. Silti mahdollisia ongelmia varten on kirjoitettu myös erillinen sovelluksen käyttöohje.

8.5.2 Rajaus

Kappaleessa 8.2.4 esitetyn hissisimulointimallin rajoitukset koskevat myös käyttöliittymää ja siinä esitettäviä asioita. Näiden lisäksi käyttöliittymä on suunniteltu vain suomenkielisille käyttäjille ja Windows-käyttöympäristölle. Pilottiprojektissa rajaus painottui helppokäyttöiseen ja toiminnalliseen käyttöliittymään, jolloin itse tietokantojen käyttö jäi vähemmälle. Jatkokehityksenä tietokantaan kerätylle datalle voidaan tehdä simulointiajojen välisiä automatisoituja vertailuja. Tässä projektissa simulointiajojen vertailussa rajoituttiin standardianalyyseista saatujen tulosteiden keskinäiseen vertailuun.

8.5.3 Rakenne ja rajapinnat

Työkalusovelluksen valikkokohtainen rakenne on esitetty kuvassa 23.



Kuva 23. Työkalusovelluksen rakenne.

Työkalusovellus on tallennettu CD-levykkeelle, mistä asennus tapahtuu koneen kovalevyllä. CD-levyyn laittaminen CD-asemaan käynnistää automaattisesti työkalusovelluksen. Ensimmäinen pääikkunan valikossa oleva vaihtoehto, jonka työkalusovellus antaa suorittaa, on ohjelman asennus. Ohjelma ei anna suorittaa muita vaihtoehtoja, sulkemisen lisäksi, ennen kuin asennus on tehty. Kuva 23 esittää asennuksen jälkeistä rakennetta.

Työkalusovelluksen rakenne jakaantuu simuloinnin näkökulmasta kolmeen osaan: lähtötietoihin, parametreihin ja tuloksiin.

Käyttäjä antaa lähtötiedot henkilöliikenteelle käyttäen ristiintaulukointia kerrosten välillä henkilöliikennevalikossa. Jokainen annettu luku siirtyy linkitetyn taulukon kautta Excelin välilehdelle, jossa se käsitellään VBA-koodin makrona ProModelin ymmärtämäksi syötedataksi. Samalla syötetty henkilöliikenne muutetaan noudattamaan mallinnettavan keskuksen mukaista jakaumaa. Tämän tekeminen on mielekästä juuri taulukkolaskennan avulla, jossa lukuja käsitellään erilaisin laskennallisoin tavoin vastaamaan hissisimuloinnin tarvetta.

Parametrejä voidaan vaihdella hissin teknisten ominaisuuksien sekä kerroskorkeuden valikoista. Näihin syötetyt luvut siirretään suoraan valikoista hissisimuloinnin koodin muuttujiksi. Syötteiden välittäminen Accesistä ProModelille on toteutettu Active-X-rajapintana, jolloin sen välityskäskyt on liitetty osaksi Accesin VBA-koodia. Itse parametrejä välittävä yhteys ohjelmien välille luodaan simuloinnin käynnistämisen yhteydessä. Tällä estetään simulointiajon aikana mahdollisen virheen syntyminen, käyttäjän mahdollisesti muuttaessa parametrejä jo seuraavaa ajoa varten.

Kolmas rakenteellinen osa, eli simuloinnista syntyvät tulokset on ohjattu erilliseen tiedostoon, josta ProModel tekee valmiiksi valitut tulostulokset käyttäjälle näytettäväksi. Tiedostosta kerätään tulokset myös Active-X-rajapinnan kautta ulkoiseen tulostietokantaan, jossa tulokset voidaan käsitellä SQL-tietokantakielellä analyysiraporttia vastaavaksi tauluksi. SQL-käsitettelyn jälkeen tietokantataulusta muodostetaan oma vakioraportti, jonka tiedot talletetaan myöhempää tarkastelua varten samaan tulostietokantaan. Analyysiraportin muotoa ja toteutusta käsitellään tarkemmin kappaleessa 8.6.

8.5.4 Visuaalisuus

Työkalusovelluksesta on visuaalisuuden peruseriaatteita noudattaen luotu tasapainoinen ja harmoninen kokonaisuus. Parametrien syöttösivujen yhtenäisellä taustaväriarinnalla on pyritty rauhoittamaan työympäristöä. Pääsivun värillinen tausta puolestaan antaa ensivaikutelman hyvin suunnitellusta tuotteesta. Jokainen tarpeellinen tieto on esitetty visuaalisesti samanlaisena, jolloin tiedot näyttävät myös käyttäjälle samanarvoisina. Kuitenkin tiedot, joita ei enää tarvita, kuten ”Asennna”-painike asennuksen jälkeen, on pyritty häivyttämään taka-alalle harmaalla värillä, jotta ne eivät turhaan vedä käyttäjien huomiota puoleensa. Tiedot on edelleen sommiteltu erillisiksi ryhmiksi omien otsikoidensa alle käyttöä selventämään.

8.5.5 Toiminnallisuuden testaaminen

Työkalusovelluksen rakentaminen jakaantui kahteen suurempaan moduuliin: simulointiin ja sovelluskehitykseen.

Simuloinnin mallintaminen jakaantui käytännössä useamman moduulien testaamiseen eli verifiointiin, jota suoritettiin kiinteänä osana mallin ohjelmoimista. Simulointimallin toimintaa verrattiin laadittuun määritelmään jatkuvasti ohjelmoinnin edetessä. Simuloinnin validointi suoritettiin tiiviissä yhteistyössä asiantuntijoiden kanssa. Toteutetun simulointimallin todettiin vastaavan riittävän tarkasti todellista hissi-liikennejärjestelmää.

Simuloinnin käytön tueksi tehty sovellus toteutettiin omana moduulinaan, keskittyen eri rajapintojen toimivuuteen. Siten integrointi eteni usein moduulitestauksen rinnalla ilman erillistä tarkastelua. Sovelluksen toiminnallisuus testattiin erikseen, kehitystyöstä erillään olleiden ihmisten avulla, lopullisella järjestelmätestauksella. Järjestelmätestauksella tarkastettiin myös käytettävyyteen liittyneet kehitysideat. Käyttöön otto eli implementointi toteutettiin osana todellista työkalusovelluskäyttöä. Tästä implementoinnista kauppakeskus Jumbolle tehdystä projektista kerrotaan myöhemmin luvussa 9.

8.6 Raportointi

Simulointituloksista kerätty datamäärä on usein suuri, ja sitä on suhteellisen vaikea lukea suoraan talletetusta tiedostosta ilman simuloinnin asiantuntemusta. Siten eri arvojen etsiminen käsittelemättömästä tulosdatasta ei anna selkeää kuvaa hissien kapasiteettien riittävydestä tai niiden suorituskyvystä. Tulosten esittämiseen tarvitaan myös tuloksiin vaikuttanut tieto käytetyistä parametreistä, jolloin tulos antaa järkevän kuvan kustakin, eri parametrein suoritetusta simulointiajosta. Tulokset antavat myös automaattisesti kirjallisen vastauksen hissien suorituskyvystä kansainvälistä käytäntöä noudattaen. Kansainvälinen käytäntö hissien suorituskyvyn mittaamisessa on esitetty taulukossa 4. Hissien lähtöväli vaikuttaa odotusaikoihin ja siten hissipalvelun laatuun. Lähtöväli on yleensä 20:n ja 40:n sekunnin väliltä, riippuen hissien lukumäärästä ja halutusta palvelutasosta.

Taulukko 4. Hissien suorituskyyky (Rakennustietosäätiö 1981).

Suorituskyyky	Lähtöväli
Tyydyttävä	33 - 40 sekunttia
Hyvä	26 - 32 sekunttia
Erittäin hyvä	< 25 sekunttia

Työkalusovelluksen standardianalyysiraportti ajetusta hissisimuloinnista käytettyine parametreineen mahtuu sellaisenaan yhdelle A4-arkille, johon on kerätty tulostiedosta tärkeimmät henkilöliikennettä koskevat tiedot. Näitä ovat henkilöiden keskimääräinen ja pisin odotusaika, hississä yhdellä kertaa enimmillään olevien ihmisten lukumäärä hisseittäin sekä hissien käyttöasteet. Kaikki kerätyt tiedot vaikuttavat hissien palvelutason tarkasteluun. Henkilöiden muodostamista jonoista saadaan myös kerroskohtainen raportointi, jossa on ilmoitettu laskennallisesti keskimääräinen ja pisin muodostunut jono. Näistä tuloksista nähdään kerroskohtaisesti mahdollisten ruuhkien syntyminen. Liitteessä 2 on esitetty esimerkki työkalusovelluksen antamasta standardianalyysistä.

9 TYÖKALUSOVELLUKSEN KÄYTTÖÖNOTTO PROJEKTISSA

Tässä luvussa kerrotaan työkalusovelluksen implementoinnista eli käyttöönotosta projektissa, joka tehtiin Vantaalla sijaitsevalle kauppakeskus Jumbolle osana suurempaa sisälogistiikan selvitystä. Kappaleissa 9.1 - 9.3 kerrotaan projektin taustasta, tavoitteista ja mitoitettavasta järjestelmästä. Kappaleessa 9.4 esitetään työkalusovelluksella tehtyjä simulointiajoja ja lopuksi kappaleissa 9.5 - 9.6 kerrotaan saaduista tuloksista ja niistä tehdyistä johtopäätöksistä.

9.1 Projektin tausta

Kauppakeskus Jumbo laajentaa tilojaan Vantaalla jo olemassa olevan ostoskeskuksen yhteyteen. Uusi laajennusosa lisää 25 000 m² jo olemassa oleviin 55 000 m² tiloihin. Suunnitelmien mukaan uuden laajennuksen rakentaminen aloitetaan vuoden 2003 toisella vuosipuoliskolla. Projektin tarkoituksena oli varmistaa kauppakeskuksen sisäisen liikenteen toimivuus, mikä käsitti sisäisen liikenteen suunnittelun kauppakeskuksen laajennusosassa. Kuvassa 24 on esitetty visualisointi kauppakeskuksen yhteyteen rakennettavasta laajennuksesta.



Kuva 24. Vasemmassa laidassa visualisointi kauppakeskus Jumbon laajennuksesta.

9.2 Ongelman määrittely ja tavoitteiden asettaminen

Projektin alussa määriteltiin tutkittava ongelma ja työkalusovelluksen avulla tutkittavat tavoitteet rakennettavan laajennuksen sisäisen liikenteen suunnittelussa. Ongel-

mana oli henkilöhissien tarvittava määrän ja koon arviointi sekä hissien riittävyyden arviointi, jos muita tasonvaihtolaitteita ei käytetä. Alustavana suunnitteluratkaisuna esitettiin kolmen hissien hissiryhmää sekä liukutasoa, mutta suunnitelmalle haettiin luotettavampaa ratkaisua hissien osalta.

Ongelman ratkaisua varten rakennetun työkalusovelluksen käytön tavoitteeksi, osana projektia, muodostui tasonvaihtolaitteiden osalta hissien suorituskyvyn ja kapasiteetin riittävyyden varmistaminen laajennusosan henkilöliikenteessä.

9.3 Mitoitettava hissijärjestelmä

Kauppakeskus Jumbon laajennusosa käsittää yhteensä 4 asiakaskerrosta, joista 2 alinta on autojen paikoitusalueita, ja 2 ylintä kerrosta on tarkoitettu liiketiloille. Tämä noudattaa samaa käytäntöä jo olemassa olevien kerrostilojen kanssa. Kauppakeskuksen laajennusosa on yhteydessä yhdyskäytävällä kauppakeskuksen vanhempaan osaan. Laajennusosaan suunnitellaan yleisiin asiakastiloihin yhtä kolmen hissien hissiryhmää ja liukuportaita, joista tässä simuloitavalla järjestelmällä tarkastellaan hissien osuutta henkilöliikenteessä.

Laajennusosan asiakasmääriä arvioitiin Jumbon nykyisten asiakasmäärien, eräiden muiden liikekeskusten toteutuneiden asiakasmäärien sekä laajennusosaa parhaiten palvelevien parkkipaikkamäärien perusteella. Mitoituksen perustaksi valittiin normaalipäivän huipputunti ja jouluruuhkan huipputunti. Kampanjamyynnin huipputunteina saattaa tätäkin suurempia asiakasmääriä tulla laajennusosaan, mutta henkilöliikennelogistiikkaa ei ole järkevä mitoittaa näitä ääritapauksia varten.

9.4 Työkalusovelluksen käyttö

Työkalusovelluksella tehtiin kaikkiaan 12 erilaista simulointiajoa. Simulointiajojen avulla arvioitiin henkilöhissijärjestelmän suorituskykyä erilaisilla henkilöliikennevirroilla ja -määrillä. Simulointiajot tehtiin tunninmittaisina ajoina, käyttäen kävijämäärissä päivän vilkkainta asiakastuntia. Ajoilla seurattiin normaalien päivien, joulunaluspäivien ja osittain myös kampanjapäivien henkilöliikennemäärien vaikutusta hissien suorituskykyyn. Ajoissa käytettiin kolmen hissien hissiryhmää, vakio pysäh-

dysaikoja ja rajoittamatonta hissien henkilökapasiteettia, jolloin mittareilla voitiin seurata hissien täyttöasteen mahdollista ylittymistä.

Ensimmäisessä simulointiajossa A1 tarkasteltiin normaalin viikonpäivän (klo 17-18) välistä asiakasliikennettä ilman liukutasoa. Tällöin kaikki ihmiset siirtyivät kerrokseen ainoastaan hissejä käyttäen. Henkilövirrat asetettiin lähtemään tasaisesti kummastakin parkkikerroksesta ja saapumaan tasaisesti kummallekin ostostasolle. Jakauma noudatti samaa käytäntöä ihmisten lähtiessä pois ostoskerroksista takaisin parkkikerrokseen.

Seuraavissa simulointiajoissa A2 - A8 lähdettiin samasta henkilövirtojen asettelusta kuin ajossa A1, mutta ihmismääriä nostettiin ajoissa asteittain. Hissiä käyttävien ihmisten lukumääriä kasvatettiin kunkin ajon alussa 5-10 % normaalipäivän asiakasmäärin verrattuna. Lopuksi simulointiajossa A8 noin puolet normaalipäivän asiakkaista käytti hissiä. Samaa normaalipäivän asiakasmäärää voitiin verrata vastaavasti prosentteina jouluruuhkan asiakasmääriin.

Simulointiajoissa A9 - A12 käytettiin puolestaan samaa, tiettyä asiakasmäärää tunnissa, mutta varioitiin henkilövirtoja eri kerrosten välillä. Myös ostoskerrostenväliset, sisäiset liikkeet olivat mukana varioitavissa ajoissa. Näillä ajoilla tarkasteltiin mallin antamien tulosten vaihteluherkkyyttä erilaisilla kerrospainotuksilla. Taulukossa 5 on esitetty eri ajoissa varioituja mahdollisia henkilövirtojen vaihteluita.

Taulukko 5. Henkilövirtojen vaihtelu kerrosten välillä.

Ajo	Parkkikerrokset		Kauppakerrokset		Sisäiset siirrot Kerros 3 <-> 4
	P 1	P 2	Kerros 3	Kerros 4	
A 9	50 %	50 %	80 %	20 %	
A 10	50 %	50 %	20 %	80 %	
A 11	80 %	20 %	20 %	80 %	5 %
A 12	50 %	50 %	70 %	30 %	10 %

Esimerkkinä taulukosta voidaan ottaa ajo A11, jossa 80% asiakasmäärästä lähtee parkkikerroksesta, tasolta P1 ja 20% tasolta P2. Parkkikerroksista lähtevät ihmiset saapuvat kauppakerrokseen suhteessa 20 % kerrokseen 3 ja 80 % kerrokseen 4. 5% asiakasmäärästä vaihtaa hissillä kerrosta sisäisesti, kolmannen ja neljännen kerroksen välillä. Henkilövirta jakautuu samalla tavalla kauppakerroksista parkkikerrokseen päin kauppakeskuksesta pois lähdettäessä.

9.5 Tulokset ja johtopäätökset

Työkalusovelluksella tehdyistä simulointiajojen tuloksista muodostettiin kappaleessa 8.6 esitetty standardianalyysiraportti, ja tuloksia esitettiin ProModelin valmiiden tulokuvaajien avulla. Myös työkalusovelluksen eri ajoista kerättyä tilastotietoa esitettiin erillisinä kuvaajina hahmottamaan ajojen välisiä eroja.

Seuraavissa kappaleissa käydään tarkemmin läpi työkalusovelluksella tehtyjen ajojen tärkeimmät tulokset ja niiden perusteella tehdyt johtopäätökset.

9.5.1 Asiakasliikenne ilman liukutasoa

Työkalusovelluksella tehdyssä ajossa A1 toteutettiin tasonvaihto pelkillä hisseillä. Tässä ajossa asiakasmäärät jopa tavallisena arkipäivänä olivat niin suuret, ettei pelkillä hisseillä ole järkevää yrittää hoitaa kaikkien henkilöiden tasonvaihtoa. Hissien käyttöasteet olivat 100 %, ja hisseille suunniteltu kapasiteetti ylittyi selvästi. Samoin keskimääräinen odotusaika nousi minuutteihin. Siten vaihtoehto toteuttaa tasonvaihto pelkällä kolmen hissien hissiryhmällä ei ole järkevä. Neljännen hissien liittäminen hissiryhmään ei tuo lähellekään riittävää apua tasonvaihtoon. Suurin osa asiakkaista tulisi siirtää tasojen välillä keskusaulaan sijoitettavalla liukutasolla.

9.5.2 Hissien suorituskky erilaisilla asiakasmäärillä

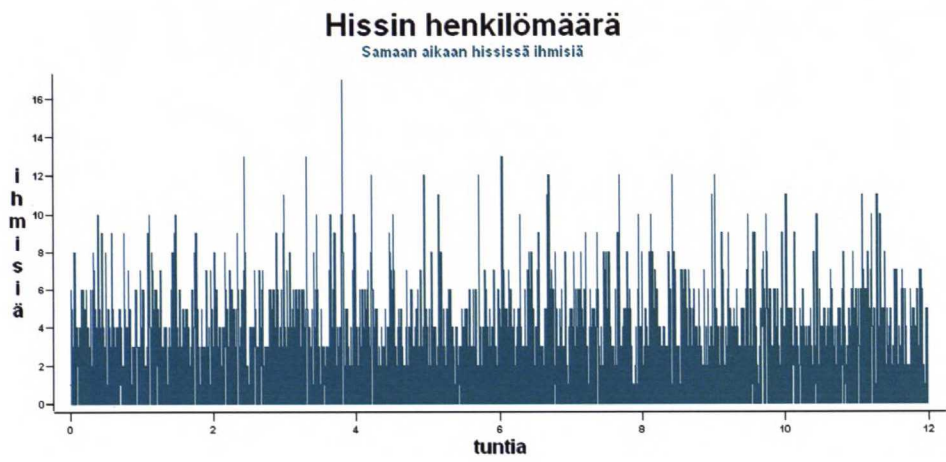
Ajojen A2 - A8 perusteella tulkittiin erilaisten asiakasmäärien vaikutusta hissille jonnottavien ihmisten odotusaikoihin ja siten hissien suorituskkyyn. Kuvassa 25 on esitetty simulointiajojen A2 - A8 tuloksista muodostettu keskimääräinen odotusaika kuvaaja eri ihmismäärillä. Tästä kuvaajasta nähdään, että kolmen ensimmäisen simulointiajon (A2 - A4) ihmismäärillä päästään erittäin hyvää suoritustasoon. Seuraavilla ajoilla (A5 - A6) päästään vielä hissien hyvään suoritustasoon, ja lopuissa ajoissa suorituskky jää tyydyttävään. Työkalusovelluksen analyysiraportti tulkitsi eri ajoista valmiiksi keskimääräisen odotusajan mukaisen suoritustason ja antoi sen kirjallisenä käyttäjälle.



Kuva 25. Kuvaaja rajoittamattomasta hissien täyttöasteesta eri ihmismäärillä.

Simulointiajoissa hissien rajoittamatonta henkilökapasiteettia käytettäessä täytyi ottaa huomioon suurten ihmismäärien kohdalla täyttöasteiden ylitys, jolloin tulokset antavat hissien suorituskyvystä liian positiivisen kuvan todelliseen kuljetuskykyyn verrattuna. Tämä näkyy kuvassa 25, jossa käyrän kasvu hidastuu käyrän loppupuolella ihmisten määrän lisääntyessä, eli odotusajat kasvavat vain hieman, vaikka asiakkaiden määrä lisääntyy koko ajan. Tässä tilanteessa käytettyjen hissien täyttöaste voi jo ylittää 100 % ja todellinen, kerralla siirtyvä ihmismäärä on hissikorin henkilömitoitusta suurempi. Näin odotusajat eivät kasva samassa suhteessa käytettäessä suuria ihmisvirtoja.

Rajoittamattoman henkilökapasiteetin käytön etuna saatiin kuitenkin mitattua maksimirajatieto henkilökuljetukseen tarvittavasta hissikoosta, ja se millä asiakasmäärillä kyseinen hissikoko saavutettiin. Kuvassa 26 on esitetty yhden ajon tuloksesta saatu kuvaaja, josta voidaan nähdä, ettei esimerkiksi 10 henkilön hissi ole riittävä. Henkilömäärän ylitys on paikoin jopa 7 ihmistä. Lisäksi hissikorin kokoa määriteltessä on otettava huomioon, siirretäänkö hisseillä myös ostoskärryjä, jotka vievät hisseistä enemmän tilaa.



Kuva 26. Aikasarja samaan aikaan hissiä käyttävien ihmisten määrästä.

Kuvassa 27 on esitetty histogrammi hissiä eripituisia aikoja odottavien ihmisten prosenttiosuuksista. Suurempi määrä hissinkäyttäjiä tunnissa aiheutti sen, että noin puolet hissimatkustajista joutuu odottamaan hissiä yli 25 sekuntia, mikä on kansainvälisen käytännön mukaan luokiteltuna erittäin hyvän palvelutason yläraja. Siten kaikkien ihmisten palvelutaso ei vastaa parasta mahdollista, mutta on silti todellisuudessa pienempien ihmismäärien käytössä riittävä jopa kahdenkin hissien hissiryhmällä.



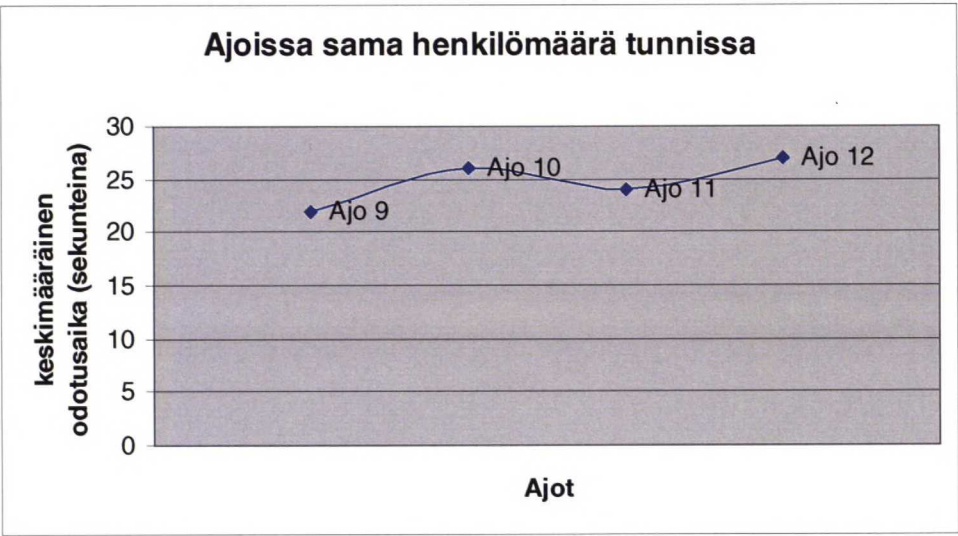
Kuva 27. Histogrammi hissiä eripituisia aikoja odottavien ihmisten prosenttiosuuksista.

Ajoista saatujen tulosten mukaan on järkevää suunnitella hissien korikoko 21 henkilölle, jolloin ihmisten lisäksi voidaan hissejä käyttää myös ostoskärryjen kuljettamiseen ilman suurempaa ruuhkaa. Käytännössä ihmisillä on myös ostoskasseja kannet-

tavanaan käsissään, jolloin hissien korikokoa ei useinkaan saada käytettyä täydellä täyttöasteella. Henkilöhissejä tullaan tarvitsemaan 2 tai 3 kappaletta hissiryhmässä. Kapasiteetin puolesta kahden hissien hissiryhmä on useasti riittävä liikutason lisäksi, mutta jos halutaan parantaa palvelutasoa, on suositeltavaa käyttää kolmen hissien hissiryhmää.

9.5.3 Suorituskyvyn vaihteluherkkyys erilaisilla kerrospainotuksilla

Ajoilla A9 - A12 tarkasteltiin työkalusovelluksen antamien tulosten vaihteluherkyyttä eri kerrospainotuksilla. Kuvassa 5 nähdään ajoista saatuja keskimääräisiä odotusaikoja, joissa kaikissa on ollut käytössä sama asiakasmäärä tunnissa.



Kuva 28. Kuvaaja vaihteluherkyydestä eri kerrospainotuksilla.

Ajossa A9 oltiin lähellä normaalia asiakaskäyttäytymistä, jolloin ensimmäiseen kauppakerrokseen tulee suhteessa enemmän ihmisiä kuin toiseen kauppakerrokseen. Tällöin saatiin vertailtavien ajojen paras tulos, joka on erittäin hyvää palvelutasoa.

Ajossa A10 käännettiin kauppakerrokseen tulleet, ajon A9 asiakasvirrat päinvastoin, jolloin toiseen kauppakerrokseen menevien ihmisten määrä oli suurempi kuin ensimmäiseen menevien. Tällaisen tilanteen voidaan kuvitella syntyvän erilaisten tapahtumien tai kampanjoiden aikana. Ajossa A10 keskimääräinen odotusaika meni yli 25 sekunnin, jolloin palvelutaso luokiteltiin vielä hyväksi palvelutasoksi. Ajoon A9 verrattuna ajon A10 keskimääräisen odotusajan piteneminen selittyy suurelta osin

hissin tekemistä pitemmistä kerrosmatkoista, joita tapahtui useammin kuin ajossa A9. Matkaan kuluva lisäaika hissillä lisää vastaavasti odottelua kerroksissa.

Ajoissa A11 ja A12 tutkittiin asiakasvirtojen vaihtelua parkkikerrosten välillä sekä kauppakerrosten välisiä, sisäisiä siirtymiä. Sisäisten siirtymien vaikutus odotusaikoja kasvattavasti ei ollut yllättävää, ottaen huomioon hissien matkaan kuluvan ajan, joka useita kertoja korkeimman kerroskorkeuden välillä liikkuesssa muodostuu merkittäväksi tekijäksi.

Erilaisten ihmisvirtojen variaatioita olisi voitu testata enemmänkin, mutta riittävä ymmärrys voitiin saavuttaa jo näillä ihmisvirtojen eri kerrospainotuksilla. Noin 5 sekunnin vaihteluväli odotusajoissa parhaimman ja heikoimman tuloksen välillä on vielä suhteellisen pieni. On huomioitava, että tulokset on mitoitettu aina asiakasmäärän huipputunnin mukaan, jolloin normaaliin vajaan käyttöön suunniteltu hissijärjestelmä antaa palvelutasonvaihteluun nähden yhtenäisimpiä tuloksia.

Täytyy muistaa, että edellä esitetyt, työkalusovellutuksen ajoista saadut tulokset ovat varmempia estimaatioita todellisuudesta, kuin erilaiset perinteisellä laskennalla tehdyt arviot, mutta toteutetuilla ajoilla ei silti pystytä jäljittelemään todellisuutta täydellisesti.

9.6 Yhteenveto johtopäätöksistä

Yhteenvetona työkalusovelluksen tuloksista arvioiduilla asiakasmäärillä voidaan todeta seuraavia asioita.

Keskusaulan liikutason lisäksi tarvitaan erityisryhmille kuten liikuntarajoitteisille ja lastenrattaiden kanssa kulkeville varten aina henkilöhisseejä. Jouluruuhkan aikaan hissejä tarvitaan myös tavallisen asiakasvirran siirtoon. Henkilöhisseejä tullaan tarvitsemaan 2 tai 3 kappaletta hissiryhmässä. 2 hissiä riittää kapasiteetin puolesta ja kolmas parantaa palvelutasoa. Hissikorin koko määräytyy myös sen mukaan, siirretäänkö hisseillä myös ostoskärryjä. Ostoskärryjen kanssa on suositeltavaa käyttää 21 henkilön korihsiiä.

Hissijärjestelmää ei ole normaalissa käyttötilanteessa järkevää mitoittaa pelkkien huipputuntien mukaan. Huipputunnit antavat silti oleellisesti hyvän käsityksen siitä, mihin hissien kapasiteetit pystyvät myös suurilla ihmismäärillä. Palvelutason laatuun voidaan silti vaikuttaa hissejä lisäämällä, jos se nähdään tarpeelliseksi.

Vaihteluherkkyys eri kerrospainotuksilla oli suhteellisen vähäistä, kun ottaa huomioon myös ihmisten hissien käytössä ilmenevät viiveet. Työkalusovelluksella annettiin ajoille sama pysähdysaika, vaikka se voi todellisuudessa vaihdella merkittävästikin, varsinkin liikuntarajoitteisilla ihmisillä.

Lyhykäisyydessään voidaan todeta, että sisäliikenteen logistiikkaselvitys puolsi alustavia suunnitteluratkaisuja, eikä suosittele alas johtavien liukutasojen poistamista. Lisäksi todettiin, että hissien määrää voidaan palvelutason kustannuksella pienentää kolmesta hissistä kahden hissien ryhmäksi, korikoon ollessa 21 henkilöä. Projektissa työkalusovellukselle asetetut tavoitteet tulivat saavutetuksi.

10 YHTEENVETO JA ARVIOINTI TYÖKALUSOVELLUKSESTA

Tässä luvussa tehdään lyhyt yhteenveto diplomityön soveltavan osan toteutuksesta. Yhteenvedossa kerrotaan työkalusovelluksen arvioinnista ja työstä saaduista hyödyistä sekä esitellään projektin aikana esiin tulleet työkalusovelluksen tulevaisuuden kehitysideat. Lopuksi vertaillaan vielä tehtyä työkalusovellusta aiemmin luvussa 7 esitettyihin vastaaviin sovelluksiin.

10.1 Työkalusovelluksen arviointi

Työkalusovellus toteutettiin diplomityön teoriaosassa esitettyjä periaatteita noudattaen. Projektille asetetut tavoitteet saavutettiin hyvässä yhteisymmärryksessä työhöni vaikuttaneiden henkilöiden kanssa niin työkalusovellusta kehittäessä kuin sen käyttöönotossa. Erityisen tärkeäksi tekijäksi työn onnistumisen kannalta osoittautui tiivis yhteistyö työkalusovelluksen tekijän ja kohdeyrityksen edustajien sekä järjestelmän asiantuntijoiden välillä. Käyttöliittymän suunnittelussa käytettyä heuristista arviointia hyödyntämällä päästiin taas hyviin laatutavoitteisiin käyttäjäystävällisyydessä. Työkalusovelluksen toteutukseen valitut tekniikat osoittautuivat tehokkaiksi ja monipuolisiksi. Samoja tekniikoita kannattaa hyödyntää myös jatkossa. Työkalusovelluksen kehittäminen onnistui tehtyjen suunnitelmien mukaisesti. Simuloinnin kehityksen aikana saadut kokemukset ja tässä työssä koottu teoria antavat jatkossa vakaamman pohjan seuraavien kehityshankkeiden työmenetelmille.

10.2 Hyödyt

Kauppakeskus Jumbon projektissa työkalusovelluksen avulla varmistettiin hissijärjestelmän toimivuus todellisessa dynaamisessa käyttöympäristössä ja saatiin aiempaa tarkempi arvio tarvittavien hissein määrästä sekä varmistettiin liukutason tarpeellisuus tasonvaihtolaitteena hissiliikenteen tukena.

Työkalusovelluksen hyödyt heijastuvat myös projektien ulkopuolelle erityyppiseen käyttöön eritasoisilla käyttäjillä. Simulointia tuntematon markkinointihenkilö voi työkalusovelluksen avulla helposti esitellä simuloinnin vahvuuksia asiakkaalle hyvin

ymmärrettävästi niin, että asiakas pääsee myös itse kokeilemaan simulointia ja näkee konkreettisesti simuloinnista saadun hyödyn.

Kokemuksen kasvaminen projektin myötä antaa tekijöilleen lisäarvoa, jota voidaan hyödyntää myöhemmin muissa projekteissa. Jo nyt simuloinnin kehittämisessä on nähtävissä, että rakennettavat käyttöliittymät tulevat tulevaisuudessa yleistymään yhä enemmän. Näin siksi, kun simuloinnin käyttö siirtyy enemmän jokapäiväiseen, operatiivisen ohjaukseen ja toiminnanohjausjärjestelmiin, aina toimitusketjun eri osapuolten yhteiseen käyttöön asti.

10.3 Työkalusovelluksen kehitysideat jatkossa

Työkalusovelluksen määrittelyssä haluttiin painottaa toiminnallisuutta, käyttäjäystävällisyyttä ja eri ohjelmien rajapintojen välistä, onnistunutta yhteyttä sekä tulosten tallentamista. Pilottiprojektiin valitut painotukset olivat luontevin tapa rakentaa, simuloinnin kehityksenä, käyttäjäystävällisempi sovellus simuloinnin käyttäjille, joilla ei ole simuloinnin asiantuntemusta.

Näistä lähtökohdista seuraavana kehitysaskelena voidaan suunnitella tallennettujen tulosten vielä pitemmälle vietyjä analysointia, diplomityöhön kuuluneen analyysiraportin rinnalle. Tämän tarve tuli esille työkalusovelluksen käyttöönotossa. Analysointiin voisi kuulua eri ajojen välinen automaattinen vertailu edellä esitettyjen kuvien 25 ja 5 mukaisesti sekä kuvina että tilastoina.

Toinen kehitysidea voisi olla työkalusovelluksen tekeminen internetin kautta toimivaksi kokonaisuudeksi, jolloin sitä voitaisiin käyttää mistä tahansa paikasta, josta saadaan yhteys WWW-sivuille. Tämä mahdollistaisi esimerkiksi tuotteistetun verkkopalvelun, jossa ajokohtaisilla maksuilla ostamalla rakennussuunnittelijat ja arkkitehtitoimistot voisivat testata hissien määrän tarvetta suunnitelluissa rakennuksissa ilman erillistä asiantuntemusta. Toinen vaihtoehto WWW-sovellukselle olisi markkinoinnin edistäminen. Yrityksen kotisivuilla voisi olla nopea ja karkea testausmahdollisuus eri rakennuksien hissimitoitukseen, josta tulokset lähetettäisiin sähköpostitse. Karkean tuloksen saaminen sähköpostissa voi lisätä käyttävän henkilön mielenkiintoa tilata tarkka hissisimulointi halutulle rakennukselle.

Kehitysideoita syntyy aina käytön yhteydessä lisää, ja näistä osa tullaan varmasti toteuttamaan myöhemmin tämän diplomityön ulkopuolisena jatkokehitysprojektina.

10.4 Työkalusovelluksen vertailu muihin sovelluksiin

Vertailtaessa työkalusovellusta kokonaisuutena vastaaviin muihin luvun 7 toteutuksiin, voidaan jossain määrin nähdä yhtäläisyyksiä kehityksessä käytetyn tekniikan valinnoissa ja sovelluksen toteutuksessa. Kuitenkin jotkut yksittäiset ratkaisut eroavat toisistaan selvästi. Työkalusovelluksen simulointiohjelma ja vastaavasti siihen käytetty rajapinta poikkeavat käytännöltään VTT:n tekemistä sovelluksista. Itse sovelluksien käytön kannalta ei voida suoraan verrata esimerkiksi käyttäjäystävällisyyttä tai sovelluksesta saatua hyötyä projektien erilaisten luonteiden vuoksi. Yhteistä kuitenkin on joustava, parametrisoitu simulointimallinnus, jolloin ei tarvita ohjelmointi- tai simulointiasiantuntijaa. Edelleen yhtenäistä sovelluksille on näkemys rakennetusta käyttöliittymästä, joka on tehty asiakkaalle edullisella ratkaisulla Windows-ympäristöön, toimisto-ohjelmia hyödyntäen.

Eroavaisuus työkalusovelluksen ja VTT:n tekemien sovellusten välillä on niiden käyttökohteissa. Indy- ja IndyNet-projektit ovat enemmän simuloinnista rakennetun operatiivisen ohjauksen työvälineitä yhdelle asiakkaalle, kun taas hissisimuloinnista on tehty suunnittelun mitoitustyökalu eri asiakkaita palvelemaan. Myöskään yhden henkilön diplomityöhön käytetyllä työpanoksella ja käytössä olevilla resursseilla ei päästä samoihin ohjelmallisiin laajennuksiin kuin VTT:llä esitetyissä, useamman henkilön projekteissa, joissa taustalla on suuremmat kehityshankkeet. Vaikka työkalusovelluksen idea olisi helppo tuotteistaa yleiseen käyttöön, on se silti tehty räätälöidysti EP-Logisticsin mitoitustyökaluksi eri projekteihin. Vertailun pohjana juuri VTT:n sovelluksiin onkin simulointimalleille tehtyt erilliset räätälöidyt käyttöliittymät, koska myös simulointimallien käyttö tulee usein olemaan, hissisimuloinnista poiketen, tapauskohtaista.

11 LÄHDELUETTELO

11.1 Kirjalliset lähteet

Banks, Jerry, John S. Carson, Barry L. Nelson, 1996. Discrete-Event System Simulation. Second Edition. United States of America: Prentice-Hall, Inc., 548 s. ISBN 0-13-217449-9

Banks, Jerry, Randall Gibson, 1997. Don't Simulate When... 10 Rules for Determining when Simulation is Not Appropriate. IIE Solutions, September, vol. 29,nro 9.

Bernold, Thomas, 1986. User Interfaces – Gateway or Bottleneck? The Netherlands: Elsevier Science Publisher B.V., 234 s. ISBN 0-444-70424-8

Boehm, B.W., 1988. A spiral Model of Software Development and Enhancement. IEEE Computer, Vol. 21, No. 5. 72 s.

Buxton, K. 2000. Simulation in the future. In: Joines, J.A. (ed). Proceedings of the 2000 Winter Simulation Conference. GA, United States of America.

Chappell, David, 1996. Understanding ActiveX and OLE. United States of America: Microsoft Press, 328 s. ISBN 1-57231-216-5

Ghezzi, Carlo, Mehdi Jazayeri, Dino Mandrioli, 1991. Fundamentals of Software Engineering. United States of America: Prentice-Hall, Inc, 573 s. ISBN 0-13-818204-3

Haikala, Ilkka, Jukka Märijärvi, 1998. 6. painos. Ohjelmistotuotanto. Helsinki: Suomen Atk-kustannus Oy, 389 s. ISBN 951-762-696-7

Hannus, Seppo, Timo Louhenkilpi , 1981. Simulointi. Espoo: OtaData, 181 s. ISBN 951-767-021-4

Harrell, Charles R., Kerim Tumay, 1995. Simulation Made Easy : a Manager's Guide. United States of America: Institute of Industrial Engineers, 311 s. ISBN 0-89806-136-9

- Hernandez, Michael J, 2000. Tietokannat – suunnittelu ja toteutus. Tomi, Kajala. Jyväskylä: Oy Edita Ab, 442 s. ISBN 951-826-137-7
- Hix, Deborah, Rex H. Hartson, 1993. Developing User Interfaces: Ensuring Usability Through Product & Process. United States of America: John Wiley & Sons, Inc., 381 s. ISBN 0-471-57813-4
- Holmström, Mikael, 1987. Relaatiotietokannat. 5. painos, Helsinki: INSKO ry. 94 s.
- Hoover, Stewart V., Ronald F. Perry, 1989. Simulation : a problem solving approach. United States of America: Addison-Wesley Publishing Company, Inc., 669 s. ISBN 0-201-16880-4
- Huotari, Heikki, 2000. Diplomityö: Tapahtumapohjainen simulointi satamalogistikan kehittämisessä. Espoo: Teknillinen Korkeakoulu, 79 s.
- Koivunen, Marja-Riitta, 1994. ActorTools: Tools for User Interface Modelling, Development, and Analysis. Doctor's Thesis, Helsinki University of Technology: Acta Polytechnica, Mathematics and Engineering Series No. 69, Helsinki 1994, 155 s. ISBN 951-666-421-0
- Koivunen, Marja-Riitta, Marko Nieminen, 1995. Ohjelmiston käytettävyys. Teoksessa: Kalimo, Anna, 1995. Graafisen käyttöliittymän suunnittelu – Opas ohjelmistojen käytettävyyteen. Espoo: Suomen ATK-kustannus Oy, 230 s. ISBN 951-762-382-8
- Korpiharju, Pekka, 2000. Tapahtumapohjainen simulointi, luentomoniste. Espoo: Otatieto, 92 s.
- Law, Averill M., David W. Kelton, 1991. Simulation modelling and analysis. Second Edition. United States of America: McGraw-Hill, Inc., 759 s. ISBN 0-07-036698-5
- Mäntylä, Martti, 1982. Tietokannan suunnittelusta. Espoo: TKK offset, 81 s. (Raportti HTKK-TKO-C2.) ISBN 951-752-506-0
- Naylor, Thomas H., 1971. Computer Simulation Experiments with Models of Economic Systems. United States of America: John Wiley & Sons, Inc., 502 s. ISBN 0-471-63070-5

Nielsen, Jakob, 1993. Usability Engineering. United States of America: Academic press, INC., 358 s. ISBN 0-12-518405-0

Nieminen, Marko, 2001. Käyttöliittymät , luentomoniste. Espoo: Edita Oy, 56s.

Pelin, Risto, 1996. Projektihallinnan käsikirja. Jyväskylä: Projektijohtaminen Oy Risto Pelin, 403 s. ISBN 951-97-430-0-6

Rakennustietosäätiö, 1981. RT-ohjetiedosto: Henkilöhistien valintaohje. RT 88-10125 (RT-kortisto). Vammalan kirjapaino Oy.

Riihiaho, Sirpa, Matti Pettersson, 1995. Käyttöliittymän suunnitteluperiaatteet. Teoksessa: Kalimo, Anna, 1995. Graafisen käyttöliittymän suunnittelu: opas ohjelmistojen käytettävyyteen. Espoo: Suomen ATK-kustannus Oy, 230 s. ISBN 951-762-382-8

Royce, Walker, 1998. Software Project Management: A Unified Framework. United States of America: Addison-Wesley. 406 s. ISBN 0201309580

Ullman, Jeffrey D, Jennifer Widom, 2002. A first course in database systems. 2nd ed. United States of America: Prentice Hall, Inc. 511 s. ISBN 0-13-035300-0

11.2 Internet-lähteet

Banks, Jerry, 2000. Introduction to Simulation. In: Joines J.(ed) Proceedings of the 2000 Winter Simulation Conference. Orlando, Florida, USA 10-13. December 2000. Viitattu 2.8.2002. Saatavissa: <http://www.informs-cs.org/wsc00papers/003.PDF>

Heilala, Juhani, Jari Montonen, Tapio Salonen, M. Hentula, 1999. Simulation Based Operations Planning In Make-to-Order Manufacturing. VTT, Manufacturing Technology. Viitattu 24.10.2002. Saatavissa: http://mango2.vtt.fi:84/val/val7/val75/valm_proj/pdf/indy9_99.pdf

Koski, Juha, 2000. SQL-ohjelmointi, opetusmoniste. Lapin Yliopisto. Viitattu 17.8.2002. Saatavissa: <http://www.urova.fi/home/mela/opetusmateriaalit/TTIE1213/SQL-ohjelmointi.pdf>

Microsoft –yhtiön kotisivut, 2002 Viitattu 12.12.2002. Saatavissa:

<http://www.microsoft.com>

Nembhard, Harriet Black, Kao Ming Shu, 2001-2002. Department of industrial Engineering, University of Wisconsin-Madison. Viitattu 20.8.2002. Saatavissa:

http://www.engr.wisc.edu/ie/faculty/nembhard_harriet/a07.pdf

Odhabi, Hamad I., Paul J.Ray, Macredie D. Robert, 1998. Developing a graphical user interface for discrete event simulation. In: Madeiros D.J., Watson E.F., Carson J.S., Manivannan M.S., eds. Proceedings of the 1998 Winter Simulation Conference. Uxbridge, Middlesex, United kindom. Viitattu 15.10.2002. Saatavissa:

<http://www.informs-cs.org/wsc98papers/058.PDF>

ProModel Corporation –yhtiön kotisivut, 2003. Viitattu 15.1.2003. Saatavissa:

<http://www.promodel.com/>

Silén, Marianne, 2002, Tietokantojen opetusmateriaali, Lapin yliopisto. Viitattu 19.8.2002. Saatavissa: <http://www.urova.fi/home/hkunta/mrissane/luennot45.doc>

Virtuaali AMK 2000, Tietokannat paikkatietojärjestelmissä. Viitattu 13.8.2002. Saatavissa:

<http://www.ncp.fi/koulutusohjelmat/metsa/paikkatietoWWW/perusteet/tietokan.htm>

11.3 Suulliset lähteet

Heilala, Juhani, 2001. Erikoistutkija. VTT Tuotteet ja tuotanto. Metallimiehenkuja 6, Espoo. Haastattelu EP-Logistics 24.10.2002.

12 LIITTEET

1. Normalisointi esimerkit.
2. Raportti työkalusovelluksen standardianalyysistä

Esimerkki tietokannan ensimmäisen normaalimuodon (1NM) käytöstä.

Alkuperäinen taulu			1NM:N mukainen taulu		
TILAUS			TILAUS		
TilausNro.	Tuote 1	Tuote 2	TilausNro.	kpl	Tuote
1001	4 ruuvia	2 sahaa	1001	4	ruuvi
1002	10 naulaa	1 vasara	1001	2	saha
			1002	10	naula
			1002	1	vasara

1NM:n edellyttämät muutokset taulussa Tilauus.

- Eritellään tuotekenttien moniatomisiet arvot omiksi yksi atomisiksi kentiksi: "Tuote" ja "kpl"
- Poistetaan toistuvat "tuote 1" ja "tuote 2" ryhmät ja liitetään ne yhdeksi

Esimerkki tietokannan toisen normaalimuodon (2NM) käytöstä.

Alkuperäinen taulu

MYYNTI					
TilausNro.	Tilaus pvm.	Asiakas	MyyjäNro.	Myyj.Nimi	Toimipaikka
1001	11.4.1977	Beach Oy	9904	Toivo	Kauhajoki
1001	23.8.1983	Honey Oy	9907	Usko	Espoo
1003	18.11.2000	Luvya Oy	9904	Onni	Helsinki

Pääavain

→

2NM:N mukaiset taulut

MYYNTI			
TilausNro.	Tilaus pvm.	Asiakas	MyyjäNro.
1001	11.4.1977	Beach Oy	9904
1001	23.8.1983	Honey Oy	9907
1003	18.11.2000	Luvya Oy	9904

Pääavain

Viiteavain

+

HENKILÖTIETO		
MyyjäNro.	Myyj.Nimi	Toimipaikka
9904	Toivo	Kauhajoki
9907	Usko	Espoo
9904	Onni	Helsinki

Pääavain

←

2NM:n edellyttämät muutokset taulussa Myynti.

- "Myyj.Nimi" tai "Toimipaikka" ei ole riippuvainen kummastakin pääavaimesta vaan ainoastaan "MyyjäNro":sta
- Siitä seuraa, että taulu Myynti on jaettava kahdeksi tauluksi: Myynti ja Henkilötieto
- Henkilötieto-tilaus "MyyjäNro":n mukana siirretyt tiedot ovat nyt yksiselitteisesti riippuvaisia uudesta pääavaimesta.
- Taulua jakamalla ei saa menettää informaatiota, joten "MyyjäNro" jää viiteavaimeksi tauluun Myynti, yhdistämään asiakkaan ja häntä palvelleeseen myyjän henkilötiedot.

Esimerkki tietokannan kolmannen normaalimuodon (3NM) käytöstä.

Alkuperäinen taulu						
TOIMITUS						
TilausNro.	Til.Osoite	Pnro	Postitoimip.	Toimittaja	LY-tunnus	Kulj.kalusto
1001	Umpikuja 6	10900	Hanko	Riski Oy	23123-123	Pakettiauto
1002	Maantie 42	96100	Rovaniemi	Rahtari Oy	42423-234	Rekka
1003	Pihakatu 1	00100	Helsinki	Lähetä Oy	51515-551	Pyörä

Pääavain

3NM:N mukaiset taulut				
TOIMITUS				
TilausNro.	Til.Osoite	Pnro	Postitoimip.	Toimittaja
1001	Umpikuja 6	10900	Hanko	Riski Oy
1002	Maantie 42	96100	Rovaniemi	Rahtari Oy
1003	Pihakatu 1	00100	Helsinki	Lähetä Oy

Pääavain

TOIMITTAJATIEDOT		
Toimittaja	LY-tunnus	Kulj.kalusto
Riski Oy	23123-123	Pakettiauto
Rahtari Oy	42423-234	Rekka
Lähetä Oy	51515-551	Pyörä
Nopea Oy	11211-112	Auto

Pääavain

3NM:n edellyttämät muutokset taulussa Toimitus.

- Avaimen ulkopuolisten kenttien "LY-tunnus" ja "Kulj.kalusto" on havaittavissa riippuvuutta "Toimittaja"-kenttään
- Hankaluutena on uusien "Toimittajien" lisääminen tietokantaan ennenkuin sillä on olemassa TilausNro.
- Taulujen jakaminen Toimitus- ja Toimittaja-tilauhin mahdollistaa uusien toimittajien lisäämisen ilman "TilausNro":a.
- Samalla tiedon tallennus ja päivittäminen vähenee moninkertaisesti.
- Tauluja jakamalla ei menetä informaatiota, kun Toimitus-tilauun yhdistetään viiteavain Toimittajatieto-tilausta

Simulointiajojen analyysiraportti

Ajon nimi **Esimerkki raportista**Päivämäärä **13.2.2003**aika **13:53:05**

Käytetyt parametrit:

Määrät:

Hissien lukumäärä: 4 kpl
 Henkilötilavuus 10 kpl
 Ostoskärrytilavuus 4 kpl

Nopeudet:

Nimellisesopeus: 2 m/s
 Kiihtyvyys: 0,8 m/s²
 Hidastavuus: 0,8 m/s²

Viiheet:

Pysähdysaika: 10 sekuntia
 Käynnistysviive 1,4 sekuntia
 Lukitsemisviive 3,4 sekuntia

Simuloinnin tulokset:

Hissi 1:n maksimi ihmismäärä	3	kpl	Hissi 1:n käyttöaste	82	%
Hissi 2:n maksimi ihmismäärä	4	kpl	Hissi 2:n käyttöaste	73	%
Hissi 3:n maksimi ihmismäärä	4	kpl	Hissi 3:n käyttöaste	65	%
Hissi 4:n maksimi ihmismäärä	4	kpl	Hissi 4:n käyttöaste	52	%

Keskimääräinen odotusaika 21 sekuntia = Erittäin hyvä suorituskyky
 Maksimi odotusaika 63 sekuntia

Keskimäärin odottavia ihmisiä kerroksittain : Maksimissaan odottavia ihmisiä kerroksittain :

Jono kerroksessa 1	1	kpl	Jono kerroksessa 1	4	kpl
Jono kerroksessa 2	1	kpl	Jono kerroksessa 2	4	kpl
Jono kerroksessa 3	0	kpl	Jono kerroksessa 3	3	kpl
Jono kerroksessa 4	0	kpl	Jono kerroksessa 4	3	kpl
Jono kerroksessa 5	0	kpl	Jono kerroksessa 5	2	kpl
Jono kerroksessa 6	0	kpl	Jono kerroksessa 6	0	kpl
Jono kerroksessa 7	0	kpl	Jono kerroksessa 7	0	kpl
Jono kerroksessa 8	0	kpl	Jono kerroksessa 8	0	kpl
Jono kerroksessa 9	0	kpl	Jono kerroksessa 9	0	kpl
Jono kerroksessa 10	0	kpl	Jono kerroksessa 10	0	kpl

